



UNIVERSITAS INDONESIA

**PEMANTAUAN KEPATUHAN OTOMATIS MELALUI ANALISIS
LOG BERBASIS AI DI MOODLE**

SKRIPSI

**YAN CHRISTOFER SILALAHI
2106752464**

**FAKULTAS FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
JUNI 2025**



UNIVERSITAS INDONESIA

**PEMANTAUAN KEPATUHAN OTOMATIS MELALUI ANALISIS
LOG BERBASIS AI DI MOODLE**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer

**YAN CHRISTOFER SILALAHI
2106752464**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
JUNI 2025**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Yan Christofer Silalahi

NPM : 2106752464

Tanda Tangan :



Tanggal : 09 Juni 2025

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Yan Christofer Silalahi
NPM : 2106752464
Program Studi : Ilmu Komputer
Judul Skripsi : Pemantauan Kepatuhan Otomatis melalui Analisis Log Berbasis AI di Moodle

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana pada Program Studi Ilmu Komputer, Fakultas Fakultas Ilmu Komputer, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing 1 : Amril Syalim, S.Kom., M.Eng., Ph.D. ()

Penguji 1 : Penguji Pertama Anda ()

Penguji 2 : Penguji Kedua Anda ()

Ditetapkan di : Depok

Tanggal : Tanggal Bulan Tahun

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Kuasa, karena di dalam penyertaan-Nya penelitian ini dapat dilaksanakan dan diselesaikan. Penulis tidak pernah lepas dari tuntunan-Nya dalam menyelesaikan laporan penelitian berjudul “Pemantauan Kepatuhan Otomatis melalui Analisis Log Berbasis AI di Moodle” ini.

Ucapan terima kasih secara khusus penulis sampaikan kepada Bapak Amril Syalim, S.Kom., M.Eng., Ph.D. yang telah membimbing proses penelitian ini dari awal hingga akhir. Segala masukan dan ilmu yang diberikan sangat berharga dan menjadi pertimbangan akademik dan karir penulis ke depannya.

Penelitian ini didukung oleh Computer Systems Lab (CSL) Fakultas Ilmu Komputer Universitas Indonesia yang telah mendukung pengembangan sistem ini. Ucapan terima kasih penulis sampaikan kepada Pak Maman dan Tim ITF yang telah mensupport akuisisi data dari awal penelitian. Dukungan teknis dan infrastruktur yang diberikan sangat membantu dalam proses pengumpulan dan analisis data log Moodle.

Terakhir, penulis mengucapkan terima kasih kepada Mama, Papa, Ivan, dan Irman yang telah memberikan dukungan tak terhingga sepanjang perjalanan akademik ini. Ucapan terima kasih juga penulis sampaikan kepada rekan-rekan “Certified Gardener” yang turut memberikan dukungan moral dan semangat dalam penyelesaian penelitian ini.

Penulis menyadari bahwa laporan Skripsi ini masih jauh dari sempurna. Oleh karena itu, apabila terdapat kesalahan atau kekurangan dalam laporan ini, penulis memohon agar kritik dan saran bisa disampaikan langsung melalui surel penulis (yan.christofer@ui.ac.id).

Depok, 09 Juni 2025



Yan Christofer Silalahi

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Yan Christofer Silalahi
NPM : 2106752464
Program Studi : Ilmu Komputer
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

Pemantauan Kepatuhan Otomatis melalui Analisis Log Berbasis AI di Moodle

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 09 Juni 2025

Yang menyatakan



(Yan Christofer Silalahi)

ABSTRAK

Nama : Yan Christofer Silalahi
Program Studi : Ilmu Komputer
Judul : Pemantauan Kepatuhan Otomatis melalui Analisis Log Berbasis AI di Moodle
Pembimbing : Amril Syalim, S.Kom., M.Eng., Ph.D.

Dalam rangka memperkuat integritas akademik pada pembelajaran era digital, penelitian ini mengembangkan sistem deteksi kecurangan yang lebih komprehensif berbasis *machine learning*. Dengan memanfaatkan data log yang kaya dan terstruktur dari Moodle, sistem mengintegrasikan beragam teknik analitik yang mencakup **deteksi anomali, clustering, dan pembelajaran terawasi** menggunakan model *advanced ensemble*. Berbagai *similarity matrix* (seperti *navigation, timing, dan answer similarity*) dikombinasikan untuk menghasilkan fitur-fitur baru yang mampu menggali pola perilaku mencurigakan. Selain itu, penerapan **gradient boosting, neural network, hingga one-class SVM dan ensemble threshold optimization** memberikan kemampuan deteksi kecurangan yang lebih akurat. Hasil evaluasi menunjukkan bahwa metode gabungan ini mampu meningkatkan sensitivitas dan spesifisitas dalam mengungkap potensi kecurangan secara proaktif, sehingga dapat menjadi landasan yang efektif bagi institusi pendidikan dalam mengurangi praktik kecurangan serta memastikan kepatuhan pengguna di platform Moodle.

Kata kunci:

Moodle, LMS, Log Aktivitas, Pembelajaran Mesin, Deteksi Anomali, Ensemble Methods, Threshold Optimization, Integritas Akademik

ABSTRACT

Name : Yan Christofer Silalahi
Study Program : Computer Science
Title : Automated Compliance Monitoring through AI-Enhanced Log Analysis on Moodle
Counselor : Amril Syalim, S.Kom., M.Eng., Ph.D.

In order to strengthen academic integrity in the digital learning era, this study develops a more comprehensive cheating detection system based on machine learning. By utilizing rich and structured log data from Moodle, our system integrates multiple analytical techniques including anomaly detection, clustering, and supervised learning through advanced ensemble models. Various similarity matrices (such as navigation, timing, and answer similarity) are combined to generate new features that uncover potentially suspicious behavior patterns. In addition, the application of gradient boosting, neural network, one-class SVM, and ensemble threshold optimization provides more accurate cheating detection capabilities. Evaluation results show that this combined method enhances both sensitivity and specificity in proactively revealing potential cheating incidents, making it an effective foundation for educational institutions to minimize dishonest practices and ensure user compliance within the Moodle platform.

Key words:

Moodle, LMS, Activity Logs, Machine Learning, Anomaly Detection, Ensemble Methods, Threshold Optimization, Academic Integrity

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN ORISINALITAS	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
LEMBAR PERSETUJUAN KARYA ILMIAH	v
ABSTRAK	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR KODE PROGRAM	xiii
1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.2.1 Pertanyaan Penelitian	3
1.2.2 Batasan Penelitian	3
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Langkah Penelitian	4
1.6 Sistematika Penulisan	5
2. STUDI LITERATUR	7
2.1 Integritas Akademik dalam Era Pembelajaran Daring	7
2.1.1 Evolusi Tantangan Integritas Akademik	7
2.1.2 Karakteristik Kecurangan dalam Lingkungan Digital	8
2.2 Pendekatan <i>Machine Learning</i> untuk Deteksi Kecurangan	8
2.2.1 Evolusi dari Sistem Berbasis Aturan ke <i>Machine Learning</i>	8
2.2.2 Teknik <i>Supervised Learning</i> untuk Deteksi Kecurangan	9
2.2.3 Deteksi Anomali dan <i>Unsupervised Learning</i>	9
2.3 Sistem Deteksi Khusus Platform Moodle	10
2.3.1 Karakteristik Data Log Moodle	10
2.3.2 Implementasi Sistem Deteksi Terintegrasi	10
2.3.3 Analisis Kesamaan dan Deteksi Kolusi	11
2.4 <i>Learning Analytics</i> dan <i>Educational Data Mining</i>	11
2.4.1 Evolusi <i>Learning Analytics</i> sebagai Disiplin	11
2.4.2 Aplikasi <i>Educational Data Mining</i> untuk Deteksi Anomali	11
2.4.3 Integrasi Perspektif Pedagogis dan Teknologis	12
2.5 Teknik Ensemble dan Optimasi Model	12
2.5.1 Pendekatan <i>Ensemble Learning</i>	12
2.5.2 Strategi Integrasi Multialgoritma	12
2.5.3 Optimasi Ambang Batas dan Hiperparameter	13
2.6 Analisis Matriks Kesamaan dan <i>Graph-Based Detection</i>	13
2.6.1 Teori Matriks Kesamaan dalam Deteksi Kolusi	13
2.6.2 Analisis Graf dan <i>Network Detection</i>	14
2.6.3 Analisis Temporal dan <i>Dynamic Networks</i>	14

2.7	Evaluasi dan Validasi Sistem Deteksi	14
2.7.1	Metrik Evaluasi dalam Konteks Akademik	14
2.7.2	Validasi Lintas Domain dan Generalisasi	15
2.7.3	Aspek Etis dan Keadilan	15
2.8	Kesenjangan Penelitian dan Peluang Pengembangan	15
2.8.1	Identifikasi Kesenjangan dalam Literatur	15
2.8.2	Peluang untuk Kontribusi Novel	16
2.9	Ringkasan	16
3.	METODE PENELITIAN	18
3.1	Desain Penelitian dan Hipotesis	20
3.1.1	Alur Tahapan Penelitian	20
3.1.2	Hipotesis Penelitian	21
3.2	Strategi Akuisisi dan Persiapan Data	22
3.3	Arsitektur Pipeline Preprocessing: Dari Log Mentah ke Fitur Terstruktur	22
3.3.1	Komponen dan Alur Kerja Pipeline	22
3.3.2	Strategi <i>Dual-Mode Processing</i>	23
3.3.3	Data Log Moodle Riil: Deskripsi (periode, jumlah event/user, fitur utama), Proses Akuisisi, Kebijakan Anonimisasi & Etika.	24
3.3.3.1	Deskripsi Data	24
3.3.3.2	Rentang Waktu dan Skala Dataset	26
3.3.3.3	Cakupan Mata Kuliah dan Pola Penggunaan	26
3.3.3.4	Proses Akuisisi dan Kebijakan Anonimisasi	27
3.3.4	Strategi Data Artifisial	28
3.4	Preprocessing Pipeline dan Feature Engineering	29
3.4.1	Pembersihan Data (Data Cleaning)	29
3.4.2	Transformasi dan Normalisasi Data	31
3.4.3	Ekstraksi Fitur dan Deteksi Outlier	31
3.4.4	<i>Checklist</i> Pra-pemrosesan	33
3.4.5	Justifikasi Ilmiah	34
3.4.6	Perancangan dan Generasi Data Artifisial	35
3.4.7	Definisi Operasional Skenario Perilaku Sintetik	35
3.4.8	Desain <i>Ground Truth</i> Artifisial	36
3.4.9	Metode Generasi Data	38
3.4.10	Implementasi Teknis	40
3.4.11	Validasi Data Artifisial	41
3.4.12	Ekstraksi dan Seleksi Fitur	43
3.4.13	Ekstraksi Fitur Dasar	44
3.4.14	Ekstraksi Fitur <i>Sequence</i> (Urutan Aktivitas)	45
3.4.15	Perhitungan <i>Similarity Features</i>	45
3.4.16	Pemeriksaan Multikolinearitas dan Seleksi Fitur Final	46
3.4.16.1	Proses Seleksi Fitur	46
3.4.16.2	Delapan Fitur Stabil Terpilih	47
3.4.17	Pra-pemrosesan Fitur untuk Kompatibilitas Model	48
3.4.18	Visualisasi dan Interpretasi Fitur	48
3.4.19	<i>Reproducibility</i> dan Dokumentasi	48

3.5	Arsitektur Model <i>Ensemble</i> untuk Deteksi Kecurangan	49
3.5.1	Desain Arsitektur <i>Multi-Model</i>	49
3.5.1.1	Komponen Model <i>Base</i> dan Perannya	49
3.5.1.2	Analisis <i>Graph Network</i> untuk Deteksi Kelompok	51
3.5.1.3	Mekanisme <i>Ensemble Integration</i>	51
3.5.2	Konfigurasi dan Optimasi Model	52
3.5.2.1	Strategi <i>Hyperparameter Tuning</i>	52
3.5.2.2	Regularisasi dan Pencegahan <i>Overfitting</i>	52
3.5.2.3	<i>Training Protocol</i> dan <i>Resource Management</i>	53
3.6	<i>Framework</i> Evaluasi Komprehensif	53
3.6.1	Evaluasi Kuantitatif pada Data Artifisial	53
3.6.1.1	Metrik Evaluasi untuk Klasifikasi <i>Imbalanced</i>	53
3.6.1.2	Protokol <i>Cross-Validation</i>	54
3.6.1.3	Analisis <i>Confusion Matrix</i> dan <i>Error Types</i>	54
3.6.2	Evaluasi Kualitatif pada Data Riil	55
3.6.2.1	Metodologi Aplikasi pada Skala Besar	55
3.6.2.2	Analisis Pola dan Validasi Domain	55
3.6.2.3	Visualisasi untuk Interpretasi	55
3.7	Kesimpulan Metodologi	56
3.7.1	Data Riil Moodle: Karakteristik dan Akuisisi	57
3.7.1.1	Profil dan Skala Dataset Riil	57
3.7.1.2	Proses Akuisisi dan Jaminan Privasi	58
3.7.1.3	Karakteristik Pola Penggunaan	58
3.7.2	Data Artifisial: Desain Terkontrol untuk <i>Ground Truth</i>	59
3.7.2.1	Justifikasi Penggunaan Data Artifisial	59
3.7.2.2	Arsitektur Generator Data Artifisial	60
3.7.2.3	Validasi Realisme Data Artifisial	60
3.8	Transformasi Data: Dari Event Log ke Representasi Fitur	61
3.8.1	Tahapan Pembersihan dan Normalisasi Data	61
3.8.2	Ekstraksi Fitur Multi-Dimensi	62
3.8.2.1	Fitur Statistik Dasar (<i>Basic Statistics</i>)	62
3.8.2.2	Fitur Pola Navigasi (<i>Navigation Patterns</i>)	63
3.8.2.3	Fitur Kemiripan Antar-Pengguna (<i>Similarity Features</i>) .	63
3.8.2.4	Fitur Anomali dan <i>Outlier</i> (<i>Anomaly Features</i>)	64
3.8.3	Analisis dan Reduksi Dimensi Fitur	64
3.8.3.1	Analisis Multikolinearitas dengan VIF	65
3.8.3.2	Delapan Fitur Final untuk Model	65
4.	EKSPERIMEN DAN ANALISIS	67
4.1	Dataset dan Konfigurasi Eksperimen	67
4.1.1	Dataset Sintesis untuk Pelatihan Model	67
4.1.1.1	Parameter Simulasi Kecurangan	68
4.1.2	Dataset Riil untuk Validasi	68
4.2	Hasil Pelatihan dan Evaluasi Model	69
4.2.1	Kinerja Model pada Data Testing	69
4.2.1.1	Analisis Confusion Matrix	69

4.2.1.2	Kurva ROC dan Precision-Recall	70
4.2.1.3	Perbandingan Kinerja Antar Model	70
4.3	Analisis Feature Importance	71
4.3.1	Fitur-Fitur yang Paling Berpengaruh	71
4.3.2	Interpretasi Fitur Berdasarkan Kategori	72
4.3.2.1	Fitur Kesamaan Navigasi (60.5%)	72
4.3.2.2	Fitur Temporal (25.4%)	72
4.3.2.3	Fitur Perilaku Penggerjaan (14.1%)	73
4.3.3	Analisis Korelasi Antar Fitur	73
4.4	Hasil Deteksi pada Data Riil	74
4.4.1	Statistik Deteksi Keseluruhan	74
4.4.2	Analisis Distribusi Probabilitas Kecurangan	74
4.4.3	Identifikasi Repeat Offenders	75
4.4.3.1	Analisis Profil Pengguna Terindikasi	75
4.4.3.2	Distribusi dan Karakteristik Repeat Offenders	75
4.4.4	Analisis Ujian dengan Tingkat Kecurangan Tinggi	76
4.5	Analisis Dampak Ukuran Dataset	77
4.5.1	Perbandingan Performa Model: 90 vs 800 Sampel	77
4.5.2	Dampak Ukuran Dataset pada Deteksi Data Riil	78
4.6	Perbandingan dengan Penelitian Terdahulu	79
4.6.1	Komparasi Performa dengan State-of-the-Art	79
4.6.2	Analisis Keunggulan Pendekatan	79
4.7	Diskusi Hasil Deteksi pada Data Riil dan Implikasi Praktis	79
4.7.1	Implikasi Praktis Deteksi pada Data Riil	80
4.7.2	Analisis Kasus Individual: Pola Navigasi, Waktu, dan Jawaban	80
4.7.3	Saran dan Insight untuk Implementasi Institusional	83
4.8	Kesimpulan	85
5. PENUTUP	87
DAFTAR REFERENSI	88

DAFTAR GAMBAR

Gambar 3.1.	Kerangka Metodologi Penelitian Deteksi Kecurangan	18
Gambar 3.2.	Arsitektur Pipeline Teknis: Alur Pemrosesan Data dari Log Mentah hingga Deteksi Kecurangan dengan Dual-Mode Processing	19
Gambar 3.3.	Alur Data dalam <i>Pipeline</i> Deteksi Kecurangan	30
Gambar 3.4.	Proses <i>Feature Engineering</i> dari <i>Raw Data</i> hingga 8 Fitur Stabil	44
Gambar 3.5.	Arsitektur <i>Ensemble</i> : Integrasi <i>Multi-Model</i> dengan <i>Graph Analysis</i> untuk Deteksi Kecurangan Komprehensif	50
Gambar 3.6.	Proses Reduksi Fitur: Dari 35 Fitur Awal menjadi 8 Fitur Stabil melalui Analisis VIF	64
Gambar 4.1.	Confusion Matrix Model Random Forest	69
Gambar 4.2.	Kurva ROC dan Precision-Recall Model Random Forest	70
Gambar 4.3.	Perbandingan Kinerja Model Machine Learning	71
Gambar 4.4.	Feature Importance Analysis Model Random Forest	71
Gambar 4.5.	Matriks Korelasi Antar Fitur Deteksi	73
Gambar 4.6.	Distribusi Probabilitas Kecurangan pada Data Riil	74
Gambar 4.7.	Analisis Distribusi Repeat Offenders	76
Gambar 4.8.	Analisis Ujian dengan Tingkat Kecurangan Tinggi	76
Gambar 4.9.	Contoh Kasus Kecurangan: User 4426 pada Quiz 3144. Gambar menunjukkan kesamaan pola navigasi, waktu penggerjaan, dan kesamaan jawaban (termasuk kesalahan identik) dengan <i>partner</i> kecurangan yang ditampilkan di bagian kanan atas	81
Gambar 4.10.	Contoh Kasus Kecurangan: User 7486 pada Quiz 8158. Terlihat kesamaan 95% dalam pola jawaban, termasuk kesalahan identik pada soal nomor 7, 12, dan 15 dengan <i>partner</i> kecurangan (ditunjukkan di bagian kanan atas)	81

DAFTAR TABEL

Tabel 3.1.	Delapan Fitur Stabil Hasil Analisis VIF	47
Tabel 3.2.	Karakteristik 8 Fitur Final Hasil Seleksi VIF	66
Tabel 4.1.	Parameter Simulasi Kecurangan dalam Dataset Sintesis	68
Tabel 4.2.	Kinerja Model pada Data Testing (120 sampel)	69
Tabel 4.3.	Lima Pengguna dengan Deteksi Kecurangan Terbanyak	75
Tabel 4.4.	Perbandingan Kinerja Model: 90 vs 800 Sampel	77
Tabel 4.5.	Perbandingan dengan Penelitian Terdahulu	79

DAFTAR KODE PROGRAM

BAB 1

PENDAHULUAN

Bab ini memaparkan fondasi penelitian yang mencakup latar belakang masalah, rumusan pertanyaan penelitian, tujuan yang ingin dicapai, serta manfaat yang diharapkan dari pengembangan sistem deteksi kecurangan akademik berbasis kecerdasan buatan.

1.1 Latar Belakang

Era digital mendorong transformasi pendidikan tinggi secara pesat, terutama sejak pandemi COVID-19. Institusi perguruan tinggi berbondong-bondong mengadopsi *Learning Management System* (LMS) seperti Moodle untuk mendukung pembelajaran daring dan pelaksanaan ujian jarak jauh Yulita et al. (2023). Digitalisasi ini membawa manfaat dalam hal fleksibilitas dan jangkauan, namun juga menimbulkan tantangan baru terhadap integritas akademik. Menjaga kejujuran akademik di lingkungan pembelajaran daring kini menjadi isu krusial, karena proses evaluasi yang berpindah ke ranah online rentan disalahgunakan untuk melakukan kecurangan Kamalov et al. (2021).

Studi menunjukkan bahwa risiko kecurangan akademik cenderung meningkat dalam konteks pembelajaran daring. Lanier menemukan tingkat kecurangan yang jauh lebih tinggi pada kelas jarak jauh dibanding kelas tatap muka tradisional Lanier (2006). Sementara itu, survei terhadap mahasiswa dan dosen di Norwegia mengidentifikasi enam modus kecurangan paling umum dalam ujian daring, seperti peniruan identitas, penggunaan bahan terlarang, dan kolaborasi tidak sah Chirumamilla et al. (2020).

Dalam konteks ini, data log aktivitas Moodle menjadi sumber informasi yang sangat berharga. Moodle secara otomatis merekam jejak interaksi pengguna secara terstruktur, mencakup waktu akses, pola navigasi, durasi penggerjaan soal, hingga kesamaan jawaban antar peserta. Analisis mendalam terhadap log ini berpotensi mengungkap berbagai pola perilaku yang mengindikasikan kecurangan, seperti kolaborasi tidak sah yang terdeteksi dari kesamaan pola navigasi dan jawaban Murdoch and House (2019), serta berbagai anomali perilaku selama pandemi Balderas and Caballero-Hernandez (2020).

Pendekatan konvensional untuk mendeteksi kecurangan umumnya mengandalkan pemeriksaan manual atau sistem berbasis aturan sederhana. Namun, metode ini memiliki

keterbatasan serius, seperti skalabilitas rendah dan kecenderungan menghasilkan *false positives* Moreno-Marcos et al. (2023).

Sebagai solusi, penelitian ini mengusulkan pendekatan berbasis *machine learning* yang mengintegrasikan beragam teknik analitik. Kerangka kerja yang dikembangkan berfokus pada model pembelajaran terawasi (*supervised learning*) seperti *Gradient Boosting*, *Random Forest*, *Support Vector Machine* (SVM), dan *Neural Network*, yang diperkuat dengan metode deteksi anomali seperti *Isolation Forest* dan *Local Outlier Factor* sebagai teknik komplementer. Pendekatan *ensemble* ini dilengkapi dengan analisis matriks kesamaan (*similarity matrices*) yang mencakup pola navigasi, waktu pengerjaan, dan jawaban mahasiswa, serta optimasi ambang batas (*threshold optimization*) untuk meningkatkan akurasi deteksi.

Implementasi sistem yang dikembangkan telah menunjukkan hasil yang sangat menjanjikan. Model *Random Forest* dan SVM mencapai akurasi 98% dengan presisi sempurna (1.00) pada dataset uji sintesis, sementara aplikasi pada data riil Moodle Fasilkom UI berhasil mengidentifikasi 131.479 percobaan ujian (29,43% dari 446.720 percobaan) dengan indikasi kecurangan berkepercayaan tinggi. Analisis *feature importance* mengungkap bahwa fitur kesamaan navigasi memberikan kontribusi paling dominan (60,5%) dalam deteksi, diikuti oleh fitur temporal (25,4%). Dengan demikian, penelitian ini berkontribusi pada pengembangan sistem deteksi kecurangan yang lebih komprehensif dan adaptif untuk platform Moodle dengan validasi empiris yang kuat.

1.2 Permasalahan

Kecurangan akademik merupakan permasalahan serius di institusi pendidikan tinggi karena dapat merusak integritas dan kualitas hasil pembelajaran. Dalam konteks pembelajaran *e-learning* menggunakan platform Moodle, potensi terjadinya kecurangan akademik semakin tinggi seiring dengan meningkatnya penggunaan ujian daring dan tugas online. Beragam bentuk kecurangan dapat terjadi, misalnya kolusi antar mahasiswa untuk berbagi jawaban, penyalahgunaan akun, atau penggunaan sumber tidak sah selama ujian.

Kompleksitas permasalahan ini menuntut pendekatan deteksi yang lebih canggih, yang tidak hanya mengandalkan satu metode, melainkan mengintegrasikan berbagai teknik analitik untuk menghasilkan deteksi yang lebih akurat dan dapat diandalkan. Diperlukan

juga kemampuan untuk menganalisis berbagai aspek perilaku pengguna secara simultan, dari pola navigasi hingga kesamaan jawaban, serta mengoptimalkan parameter deteksi untuk meminimalkan kesalahan klasifikasi.

1.2.1 Pertanyaan Penelitian

Berdasarkan permasalahan yang telah diuraikan, penelitian ini berusaha menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana mengembangkan pendekatan berbasis pembelajaran mesin yang efektif untuk mendeteksi potensi kecurangan akademik dalam pembelajaran daring menggunakan data log aktivitas Moodle?
2. Sejauh mana integrasi berbagai teknik analisis data dapat meningkatkan akurasi dan reliabilitas deteksi perilaku mencurigakan dalam konteks pembelajaran daring?
3. Bagaimana karakteristik dan pola perilaku pengguna yang teridentifikasi dari hasil analisis dapat memberikan wawasan untuk meningkatkan integritas akademik dalam pembelajaran daring?

1.2.2 Batasan Penelitian

Untuk memastikan penelitian ini tetap terfokus dan terarah, beberapa batasan dan ruang lingkup berikut diterapkan:

1. Lingkup Data: Data yang digunakan dalam penelitian ini dibatasi pada log aktivitas pengguna dari platform Moodle di lingkungan Fasilkom UI, dengan model dilatih menggunakan dataset artifisial berjumlah 800 sampel yang karakteristiknya divalidasi terhadap data riil, dan kemudian diterapkan pada 446.720 percobaan ujian riil dari data log Fasilkom UI untuk analisis.
2. Jenis Kecurangan: Deteksi difokuskan pada pola perilaku mencurigakan yang tercermin dalam log aktivitas, matriks kesamaan, dan interaksi antar pengguna.
3. Metode dan Algoritma: Pendekatan utama menggunakan model pembelajaran terawasi dengan *ensemble* (*Random Forest*, *SVM*, *Neural Network*, *Gradient Boosting*), didukung metode deteksi anomali sebagai komplemen.
4. Mode Implementasi: Sistem deteksi diimplementasikan dalam modus *offline* untuk analisis retrospektif.
5. Evaluasi: Kinerja sistem dievaluasi menggunakan metrik standar seperti presisi, *recall*,

skor F1, dan *Area Under Curve* (AUC) ROC.

1.3 Tujuan Penelitian

Penelitian ini memiliki tujuan utama untuk mengembangkan sistem deteksi kecurangan yang komprehensif berbasis analisis log aktivitas Moodle. Secara terperinci, tujuan penelitian ini adalah:

1. Merancang dan mengimplementasikan kerangka kerja deteksi yang mengintegrasikan model pembelajaran terawasi dengan *ensemble*, didukung metode deteksi anomali, analisis matriks kesamaan, dan optimasi ambang batas.
2. Mengembangkan dan mengevaluasi fitur-fitur baru berbasis matriks kesamaan untuk meningkatkan akurasi deteksi.
3. Melakukan pengujian menyeluruh terhadap kinerja sistem menggunakan data log Moodle Fasilkom UI.
4. Menganalisis dan menginterpretasikan pola-pola perilaku mencurigakan yang terdeteksi untuk mendukung upaya pencegahan kecurangan.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat baik secara teoretis maupun praktis:

1. Manfaat Teoretis:
 - (a) Kontribusi pada pengembangan metode deteksi kecurangan berbasis *ensemble*.
 - (b) Pemahaman baru tentang efektivitas matriks kesamaan dalam analisis perilaku.
 - (c) Landasan metodologis untuk penelitian lanjutan.
2. Manfaat Praktis:
 - (a) Sistem deteksi dini yang lebih akurat untuk institusi pendidikan.
 - (b) Dukungan objektif untuk pengambilan keputusan terkait integritas akademik.
 - (c) Peningkatan efektivitas monitoring pembelajaran daring.
 - (d) Dasar pengembangan sistem deteksi *real-time* di masa depan.

1.5 Langkah Penelitian

Berikut ini adalah langkah penelitian yang dilakukan:

1. Tinjauan Literatur

Mengkaji teori dan penelitian terkait deteksi kecurangan, metode *ensemble*, dan analisis matriks kesamaan dalam konteks pembelajaran daring.

2. Pengumpulan dan Pengolahan Data

Mengumpulkan log aktivitas Moodle, melakukan pembersihan data, dan mengekstraksi fitur-fitur relevan termasuk matriks kesamaan.

3. Pengembangan Sistem

Mengimplementasikan kerangka kerja deteksi yang mengintegrasikan model pembelajaran terawasi (*Gradient Boosting, Random Forest, Neural Network*) sebagai komponen utama, diperkaya dengan metode deteksi anomali, analisis matriks kesamaan, dan optimasi ambang batas.

4. Evaluasi dan Analisis

Menguji kinerja sistem menggunakan metrik standar, menganalisis pola-pola yang terdeteksi, dan menginterpretasikan implikasinya.

5. Penarikan Kesimpulan

Menyimpulkan efektivitas pendekatan yang diusulkan dan merumuskan rekomendasi untuk pengembangan sistem dan penelitian lanjutan.

1.6 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini adalah sebagai berikut:

- **Bab 1 Pendahuluan**

Berisi latar belakang penelitian, perumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian, langkah-langkah penelitian, serta sistematika penulisan.

- **Bab 2 Studi Literatur**

Mengkaji konsep-konsep fundamental tentang integritas akademik dalam pembelajaran daring, teknik-teknik pembelajaran mesin untuk deteksi anomali, serta penelitian-penelitian terkait dalam bidang analisis perilaku pengguna sistem pembelajaran daring.

- **Bab 3 Metodologi Penelitian**

Menjelaskan pendekatan metodologis yang digunakan, termasuk desain sistem deteksi, proses pengolahan data, pemilihan dan integrasi metode analisis, serta kerangka evaluasi yang diterapkan.

- **Bab 4 Eksperimen dan Analisis**

Memaparkan hasil implementasi sistem, analisis kinerja model berdasarkan berbagai

metrik evaluasi, serta interpretasi temuan dari aplikasi sistem pada data riil Moodle Fasilkom UI.

- **Bab 5 Kesimpulan**

Menyajikan kesimpulan penelitian, keterkaitan dengan tujuan dan pertanyaan penelitian, keterbatasan yang ditemui, serta rekomendasi untuk pengembangan dan penelitian lanjutan.

BAB 2

STUDI LITERATUR

Bab ini menyajikan tinjauan pustaka yang komprehensif mengenai landasan teoretis dan penelitian terkait sistem deteksi kecurangan akademik berbasis kecerdasan buatan. Pembahasan mencakup evolusi masalah integritas akademik dalam pembelajaran daring, perkembangan teknik *machine learning* untuk deteksi anomali, serta aplikasi spesifik dalam lingkungan *Learning Management System* (LMS) seperti *Moodle*.

2.1 Integritas Akademik dalam Era Pembelajaran Daring

2.1.1 Evolusi Tantangan Integritas Akademik

Integritas akademik telah menjadi perhatian fundamental dalam dunia pendidikan sejak lama, namun transformasi digital pendidikan telah mengubah secara signifikan lanskap dan karakteristik permasalahan ini. Lanier Lanier (2006) dalam penelitiannya yang menjadi rujukan penting, menemukan bahwa tingkat kecurangan dalam pembelajaran jarak jauh cenderung lebih tinggi dibandingkan dengan kelas tatap muka tradisional. Temuan ini menjadi dasar pemahaman bahwa lingkungan pembelajaran daring memerlukan pendekatan pemantauan yang berbeda dan lebih komprehensif.

Chirumamilla dkk. Chirumamilla et al. (2020) melalui survei terhadap mahasiswa dan dosen di Norwegia, mengidentifikasi enam modus kecurangan paling umum dalam ujian daring: (1) peniruan identitas (*identity theft*), (2) penggunaan bahan bantuan terlarang, (3) kolaborasi tidak sah antarpeserta, (4) penggunaan perangkat komunikasi selama ujian, (5) akses ke sumber eksternal tanpa izin, dan (6) manipulasi waktu pengerjaan. Klasifikasi ini memberikan kerangka pemahaman yang sistematis tentang berbagai bentuk pelanggaran yang perlu dideteksi oleh sistem otomatis.

Penelitian terbaru menunjukkan bahwa pandemi COVID-19 telah mempercepat adopsi pembelajaran daring sekaligus meningkatkan kompleksitas permasalahan integritas akademik. Yulita dkk. Yulita et al. (2023) mengamati peningkatan signifikan dalam kecanggihan metode kecurangan, termasuk penggunaan teknologi untuk memfasilitasi kolusi dan berbagi informasi secara real-time selama ujian berlangsung.

2.1.2 Karakteristik Kecurangan dalam Lingkungan Digital

Lingkungan pembelajaran digital memiliki karakteristik unik yang membedakannya dari pengaturan tradisional. Murdoch dan House Murdoch and House (2019) mengidentifikasi fenomena "*ghost in the shell*", yaitu perpaduan antara kecurangan berbasis kontrak (*contract cheating*) dengan peniruan identitas daring. Fenomena ini menunjukkan evolusi kecurangan dari tindakan individual menjadi operasi yang lebih terorganisasi dan teknologis.

Balderas dan Caballero-Hernández Balderas and Caballero-Hernández (2020) dalam analisis mereka terhadap rekam jejak pembelajaran selama pandemi, menemukan bahwa pola perilaku mencurigakan dapat diidentifikasi melalui analisis temporal dan spasial aktivitas mahasiswa. Temuan ini memperkuat argumen bahwa data log aktivitas mengandung informasi yang kaya untuk deteksi kecurangan, asalkan dianalisis dengan metode yang tepat.

2.2 Pendekatan *Machine Learning* untuk Deteksi Kecurangan

2.2.1 Evolusi dari Sistem Berbasis Aturan ke *Machine Learning*

Pendekatan tradisional untuk deteksi kecurangan akademik umumnya mengandalkan sistem berbasis aturan (*rule-based systems*) yang menggunakan ambang batas statis untuk mengidentifikasi perilaku mencurigakan. Huda dkk. Huda et al. (2020) mengidentifikasi beberapa keterbatasan fundamental dari pendekatan ini: (1) rendahnya akurasi dalam menangani pola perilaku yang kompleks, (2) tingginya tingkat *false positive* yang mengakibatkan banyak mahasiswa normal yang salah dituduh, (3) ketidakmampuan untuk beradaptasi dengan modus kecurangan yang berkembang, dan (4) kesulitan dalam menangani variasi kontekstual antarmata kuliah atau institusi.

Sebagai respons terhadap keterbatasan ini, penelitian modern beralih ke pendekatan berbasis *machine learning* yang menawarkan kemampuan adaptif dan akurasi yang lebih tinggi. Kamalov dkk. Kamalov et al. (2021) menunjukkan bahwa model pembelajaran mesin dapat mencapai akurasi deteksi hingga 94% dalam mengidentifikasi kecurangan ujian, jauh melampaui kinerja sistem berbasis aturan tradisional.

2.2.2 Teknik *Supervised Learning* untuk Deteksi Kecurangan

Pendekatan pembelajaran terawasi (*supervised learning*) telah menjadi metode dominan dalam deteksi kecurangan akademik karena kemampuannya dalam mempelajari pola dari data berlabel. Zhou dan Jiao Zhou and Jiao (2022) dalam penelitian mereka tentang augmentasi data untuk deteksi kecurangan dalam asesmen skala besar, mendemonstrasikan efektivitas berbagai algoritma pembelajaran terawasi, termasuk *Random Forest*, *Support Vector Machine*, dan *Gradient Boosting*.

Alsabhan Alsabhan (2023) mengembangkan pendekatan hibrida yang mengintegrasikan *Long Short-Term Memory* (LSTM) dengan teknik pembelajaran mesin tradisional untuk mendeteksi kecurangan mahasiswa di perguruan tinggi. Penelitian ini menunjukkan bahwa kombinasi neural network dengan algoritma ensemble dapat meningkatkan akurasi deteksi hingga 96,8%, dengan kemampuan khusus dalam menangkap pola temporal yang kompleks dalam perilaku mahasiswa.

Chang dan Chang Chang and Chang (2023) melakukan studi komprehensif tentang deteksi kolusi dalam ujian, dengan fokus pada teknik pembelajaran mesin dan representasi fitur. Mereka menemukan bahwa *feature engineering* yang tepat, terutama yang berkaitan dengan matriks kesamaan dan analisis graf, dapat secara signifikan meningkatkan kemampuan deteksi kolaborasi tidak sah antarpeserta ujian.

2.2.3 Deteksi Anomali dan *Unsupervised Learning*

Meskipun pembelajaran terawasi menunjukkan kinerja yang baik, ketersediaan data berlabel yang berkualitas seringkali menjadi kendala dalam implementasi praktis. Sebagai alternatif, pendekatan deteksi anomali berbasis *unsupervised learning* menawarkan solusi yang menjanjikan. Cen dkk. Cen et al. (2020) mengembangkan kerangka kerja untuk deteksi anomali tanpa pengawasan dalam sistem *e-learning*, yang mampu mengidentifikasi pola perilaku yang tidak biasa tanpa memerlukan data berlabel sebelumnya.

Alexandron dkk. Alexandron et al. (2019) mengusulkan metode deteksi anomali tujuan umum untuk mengidentifikasi kecurangan dalam *Massive Open Online Courses* (MOOC). Penelitian mereka menunjukkan bahwa teknik deteksi anomali seperti *Isolation Forest* dan *Local Outlier Factor* dapat efektif dalam mengidentifikasi pola perilaku yang menyimpang dari norma, dengan tingkat presisi yang dapat diterima untuk implementasi praktis.

2.3 Sistem Deteksi Khusus Platform Moodle

2.3.1 Karakteristik Data Log Moodle

Moodle sebagai salah satu LMS yang paling banyak digunakan di dunia, menyediakan sistem pencatatan log yang komprehensif dan terstruktur. Mazza dan Dimitrova Mazza and Dimitrova (2004) dalam penelitian pionir mereka, menjelaskan bahwa log aktivitas Moodle mengandung informasi detail tentang setiap interaksi pengguna dengan platform, termasuk timestamp, tipe aktivitas, durasi, dan konteks akademik.

Data log Moodle memiliki beberapa karakteristik unik yang membuatnya sangat cocok untuk analisis *machine learning*: (1) granularitas tinggi dalam perekaman aktivitas, (2) konsistensi format data lintas berbagai modul, (3) integrasi dengan konteks pembelajaran yang memungkinkan analisis berbasis mata kuliah, dan (4) kemampuan pelacakan yang mencakup tidak hanya aktivitas ujian tetapi juga pola belajar secara keseluruhan.

2.3.2 Implementasi Sistem Deteksi Terintegrasi

Shatnawi dkk. Shatnawi et al. (2024) mengembangkan sistem deteksi kecurangan ujian elektronik yang terintegrasi langsung dengan platform Moodle LMS. Sistem ini menggunakan pendekatan pembelajaran mesin dengan metode statistik yang mampu mencapai akurasi 100% dalam mendeteksi berbagai jenis anomali, termasuk deteksi waktu respons yang tidak wajar, pola navigasi mencurigakan, dan aktivitas yang tidak konsisten dengan perilaku normal mahasiswa.

Moreno-Marcos dkk. Moreno-Marcos et al. (2023) mengembangkan Statoodle, sebuah alat *learning analytics* yang diintegrasikan langsung dengan Moodle untuk menganalisis aksi mahasiswa dan mencegah kecurangan. Sistem ini menggunakan pendekatan pemantauan waktu nyata yang dapat memberikan peringatan dini kepada pengawas ujian ketika terdeteksi aktivitas mencurigakan.

Pendekatan terintegrasi ini menawarkan beberapa keuntungan: (1) akses langsung ke data log tanpa perlu ekspor manual, (2) kemampuan pemantauan waktu nyata, (3) integrasi dengan alur kerja yang ada di institusi pendidikan, dan (4) kemudahan dalam implementasi tindakan preventif atau responsif.

2.3.3 Analisis Kesamaan dan Deteksi Kolusi

Salah satu kekuatan utama platform Moodle adalah kemampuannya dalam menyediakan data yang memungkinkan analisis kesamaan antarpeserta. Chang dan Chang Chang and Chang (2023) menunjukkan bahwa analisis matriks kesamaan berbasis jawaban, pola navigasi, dan waktu dapat secara efektif mengidentifikasi kolusi antarmahasiswa.

Teknik analisis graf juga dapat diterapkan pada data Moodle untuk mengidentifikasi kluster mahasiswa yang menunjukkan pola perilaku yang serupa secara tidak wajar. Pendekatan ini tidak hanya dapat mendeteksi kecurangan individual, tetapi juga mengungkap jaringan kolaborasi yang lebih luas.

2.4 *Learning Analytics* dan *Educational Data Mining*

2.4.1 Evolusi *Learning Analytics* sebagai Disiplin

Learning analytics telah berkembang menjadi disiplin yang matang dalam dekade terakhir, dengan fokus pada penggunaan data pendidikan untuk meningkatkan proses dan hasil pembelajaran. Siemens dan Long (2011) mendefinisikan *learning analytics* sebagai pengukuran, pengumpulan, analisis, dan pelaporan data tentang pelajar dan konteks mereka, dengan tujuan memahami dan mengoptimalkan pembelajaran serta lingkungan tempat pembelajaran tersebut terjadi.

Dalam konteks deteksi kecurangan, *learning analytics* menyediakan kerangka metodologis dan teknologis yang komprehensif. Ferguson (2012) mengidentifikasi bahwa pendekatan *learning analytics* tidak hanya fokus pada deteksi masalah, tetapi juga pada pemahaman mendalam tentang pola perilaku belajar yang dapat membantu dalam pencegahan proaktif.

2.4.2 Aplikasi *Educational Data Mining* untuk Deteksi Anomali

Educational Data Mining (EDM) sebagai subbidang dari *learning analytics* menyediakan teknik-teknik khusus untuk ekstraksi pola dari data pendidikan. Romero dan Ventura (2020) dalam ulasan komprehensif mereka, mengidentifikasi bahwa teknik EDM telah berkembang dari analisis deskriptif sederhana menjadi model prediktif yang kompleks.

Dalam konteks deteksi kecurangan, EDM menawarkan beberapa teknik yang relevan: (1) *sequence mining* untuk menganalisis pola navigasi, (2) *clustering* untuk mengidentifikasi

grup mahasiswa dengan perilaku serupa, (3) *association rule mining* untuk menemukan hubungan antaraktivitas, dan (4) *classification* untuk membedakan perilaku normal dan mencurigakan.

2.4.3 Integrasi Perspektif Pedagogis dan Teknologis

Aspek penting dalam pengembangan sistem deteksi kecurangan adalah integrasi antara perspektif pedagogis dan teknologis. Gašević dkk. (2015) menekankan bahwa sistem *learning analytics* yang efektif harus mempertimbangkan tidak hanya aspek teknis deteksi, tetapi juga implikasi pedagogis dan etis dari implementasi sistem tersebut.

Dalam konteks deteksi kecurangan, hal ini berarti sistem harus dirancang tidak hanya untuk mengidentifikasi pelanggaran, tetapi juga untuk mendukung proses pembelajaran yang adil dan mendorong integritas akademik melalui pendekatan yang konstruktif daripada sekadar bersifat hukuman.

2.5 Teknik Ensemble dan Optimasi Model

2.5.1 Pendekatan *Ensemble Learning*

Ensemble learning telah terbukti sebagai salah satu pendekatan paling efektif dalam meningkatkan akurasi dan ketahanan model *machine learning*. Dalam konteks deteksi kecurangan akademik, teknik ensemble menawarkan keuntungan khusus karena kemampuannya dalam menggabungkan kekuatan berbagai algoritma untuk menangani kompleksitas dan variasi pola kecurangan.

Zhou (2012) dalam "Ensemble Methods: Foundations and Algorithms" menjelaskan bahwa kekuatan ensemble terletak pada prinsip "diversity and accuracy", yaitu kombinasi model yang beragam namun akurat dapat menghasilkan kinerja yang superior dibandingkan model individual. Dalam konteks deteksi kecurangan, keberagaman ini sangat penting karena berbagai jenis kecurangan mungkin lebih baik dideteksi oleh algoritma yang berbeda.

2.5.2 Strategi Integrasi Multialgoritma

Penelitian terkini menunjukkan bahwa integrasi strategis antara model pembelajaran terawasi dengan teknik deteksi anomali dapat menghasilkan sistem yang lebih kuat. Nadeem

dkk. Nadeem et al. (2024) dalam penelitian mereka tentang teknik pembelajaran mesin canggih untuk deteksi penipuan keuangan, menunjukkan bahwa kombinasi supervised dan unsupervised learning dapat mengatasi keterbatasan masing-masing pendekatan: supervised learning memberikan akurasi tinggi pada pola yang dikenal, sementara unsupervised learning dapat mendeteksi anomali baru yang belum pernah ditemui sebelumnya.

Dalam implementasi praktis, strategi ensemble dapat mencakup: (1) *voting classifiers* yang menggabungkan prediksi beberapa model, (2) *stacking* yang menggunakan meta-learner untuk mengoptimalkan kombinasi, (3) *bagging* untuk mengurangi varians, dan (4) *boosting* untuk mengurangi bias.

2.5.3 Optimasi Ambang Batas dan Hiperparameter

Optimasi ambang batas merupakan aspek kritis dalam sistem deteksi kecurangan karena pertukaran antara false positive dan false negative memiliki implikasi praktis yang signifikan. Ambang batas yang terlalu rendah akan menghasilkan banyak false positive yang dapat merugikan mahasiswa yang tidak bersalah, sementara ambang batas yang terlalu tinggi dapat membiarkan kecurangan lolos dari deteksi.

Niu dkk. Niu et al. (2025) dalam ulasan sistematis tentang metodologi pembelajaran mesin yang efektif menunjukkan bahwa optimasi ambang batas sebaiknya dilakukan dengan mempertimbangkan cost-sensitive learning, yaitu *cost* dari berbagai jenis kesalahan diperhitungkan secara eksplisit. Dalam konteks akademik, *cost* dari false positive (menuduh mahasiswa yang tidak bersalah) mungkin berbeda dengan *cost* dari false negative (membiarkan pelanggar akademik lolos).

2.6 Analisis Matriks Kesamaan dan *Graph-Based Detection*

2.6.1 Teori Matriks Kesamaan dalam Deteksi Kolusi

Analisis matriks kesamaan telah menjadi teknik fundamental dalam deteksi kolaborasi tidak sah dalam konteks akademik. Konsep ini didasarkan pada premis bahwa mahasiswa yang melakukan kolusi akan menunjukkan pola perilaku yang tidak natural serupa, baik dalam hal jawaban, pola navigasi, maupun waktu.

Ukuran kesamaan yang umum digunakan dalam konteks ini meliputi: (1) *Cosine similarity*

ity untuk mengukur kemiripan vektor fitur, (2) *Jaccard similarity* untuk data kategorikal, (3) *Pearson correlation* untuk hubungan linear, dan (4) *Edit distance* untuk data sekuen-sial. Pemilihan ukuran yang tepat sangat bergantung pada jenis data dan karakteristik kecurangan yang ingin dideteksi.

2.6.2 Analisis Graf dan *Network Detection*

Pendekatan berbasis graf menawarkan perspektif yang kuat untuk memahami pola kolaborasi dalam skala yang lebih besar. Dalam representasi graf, mahasiswa dapat dimodelkan sebagai simpul, sementara sisi merepresentasikan tingkat kesamaan atau kolaborasi yang dicurigai.

Teknik analisis graf yang relevan meliputi: (1) *community detection* untuk mengidentifikasi kluster mahasiswa yang berkolaborasi, (2) *centrality measures* untuk mengidentifikasi individu yang menjadi pusat dalam jaringan kolaborasi, (3) *clustering coefficient* untuk mengukur tingkat interkoneksi, dan (4) *modularity analysis* untuk memvalidasi struktur komunitas.

2.6.3 Analisis Temporal dan *Dynamic Networks*

Aspek temporal dalam analisis kesamaan memberikan dimensi tambahan yang penting. Kecurangan seringkali menunjukkan pola temporal yang karakteristik, seperti pengiriman jawaban secara bersamaan, pola navigasi yang sinkron, atau perubahan jawaban yang terkoordinasi.

Analisis jaringan dinamis dapat mengungkap pola kolaborasi yang berkembang selama ujian berlangsung, memberikan wawasan yang tidak dapat diperoleh dari analisis statis. Hal ini sangat relevan untuk ujian yang berlangsung dalam periode waktu yang diperpanjang atau untuk menganalisis pola kecurangan lintas beberapa sesi.

2.7 Evaluasi dan Validasi Sistem Deteksi

2.7.1 Metrik Evaluasi dalam Konteks Akademik

Evaluasi sistem deteksi kecurangan memerlukan pertimbangan khusus karena karakteristik unik dari domain akademik. Metrik evaluasi standar seperti akurasi, presisi, recall, dan F1-score tetap relevan, namun interpretasi dan prioritas mereka harus disesuaikan dengan

konteks.

Dalam pengaturan akademik, false positive (menuduh mahasiswa yang tidak bersalah) memiliki implikasi yang sangat serius, termasuk kerusakan reputasi, stres psikologis, dan konsekuensi hukum yang potensial. Oleh karena itu, presisi menjadi metrik yang sangat kritis. Di sisi lain, false negative (membiarkan pelanggar akademik lolos) dapat merusak keadilan sistem evaluasi dan mengancam integritas akademik secara keseluruhan.

2.7.2 Validasi Lintas Domain dan Generalisasi

Salah satu tantangan utama dalam pengembangan sistem deteksi kecurangan adalah memastikan generalisasi lintas mata kuliah, institusi, dan konteks yang berbeda. Model yang dilatih pada satu set data mungkin tidak berkinerja baik pada konteks yang berbeda karena variasi dalam perilaku mahasiswa, struktur mata kuliah, atau format ujian.

Strategi validasi silang yang kuat perlu mempertimbangkan tidak hanya pembagian acak, tetapi juga stratifikasi berdasarkan jenis mata kuliah, tingkat mahasiswa, atau faktor temporal. Hal ini penting untuk memastikan bahwa model dapat menggeneralisasi ke situasi dunia nyata yang beragam.

2.7.3 Aspek Etis dan Keadilan

Implementasi sistem deteksi kecurangan otomatis menimbulkan berbagai pertanyaan etis yang perlu dipertimbangkan secara serius. Keadilan algoritma menjadi isu yang kritis, terutama terkait dengan bias potensial terhadap kelompok mahasiswa tertentu.

Aspek etis yang perlu dipertimbangkan meliputi: (1) transparansi dalam proses pengambilan keputusan, (2) dapat dijelaskannya prediksi yang dihasilkan, (3) keadilan lintas kelompok demografis yang berbeda, (4) perlindungan privasi dalam penanganan data, dan (5) pengawasan manusia dalam proses pengambilan keputusan final.

2.8 Kesenjangan Penelitian dan Peluang Pengembangan

2.8.1 Identifikasi Kesenjangan dalam Literatur

Meskipun telah terdapat banyak penelitian dalam bidang deteksi kecurangan akademik, beberapa kesenjangan masih dapat diidentifikasi: (1) kurangnya penelitian yang fokus

pada integrasi komprehensif antara beberapa teknik, (2) penelitian terbatas pada optimasi ambang batas untuk meminimalkan false positive dalam konteks asesmen akademik berisiko tinggi, (3) perhatian yang tidak memadai pada dinamika temporal dan evolusi pola kecurangan, dan (4) kurangnya standar pembanding untuk evaluasi komparatif.

2.8.2 Peluang untuk Kontribusi Novel

Penelitian ini memiliki peluang untuk memberikan kontribusi dalam beberapa area: (1) pengembangan kerangka kerja ensemble yang mengintegrasikan supervised learning, deteksi anomali, dan analisis kesamaan secara optimal, (2) inovasi dalam feature engineering berbasis matriks kesamaan yang dapat menangkap pola kolaborasi yang kompleks, (3) pengembangan optimasi ambang batas yang sadar konteks yang dapat beradaptasi dengan pengaturan akademik yang berbeda, dan (4) penciptaan kerangka kerja evaluasi komprehensif yang mempertimbangkan kinerja teknis dan implikasi praktis.

2.9 Ringkasan

Tinjauan pustaka ini menunjukkan bahwa deteksi kecurangan akademik dalam pembelajaran daring telah berkembang dari sistem berbasis aturan sederhana menjadi implementasi *machine learning* yang canggih. Integrasi berbagai teknik analitik, termasuk *supervised learning*, deteksi anomali, analisis matriks kesamaan, dan pendekatan ensemble, menawarkan potensi untuk mengembangkan sistem deteksi yang lebih akurat dan kuat.

Penelitian-penelitian yang diulas menunjukkan bahwa tidak ada teknik tunggal yang optimal untuk semua jenis perilaku kecurangan. Sebaliknya, pendekatan terintegrasi yang menggabungkan kekuatan algoritma yang berbeda sambil memitigasi kelemahan masing-masing menunjukkan hasil yang paling menjanjikan.

Platform Moodle, dengan sistem pencatatan log yang komprehensif dan kemampuan integrasi yang baik, menyediakan lingkungan yang ideal untuk implementasi sistem deteksi canggih. Data log yang kaya dan terstruktur memungkinkan ekstraksi fitur beragam yang dapat menangkap berbagai aspek perilaku mahasiswa.

Namun, implementasi praktis sistem deteksi otomatis juga menimbulkan tantangan terkait keadilan, etika, dan penerapan praktis. Oleh karena itu, pengembangan sistem yang tidak

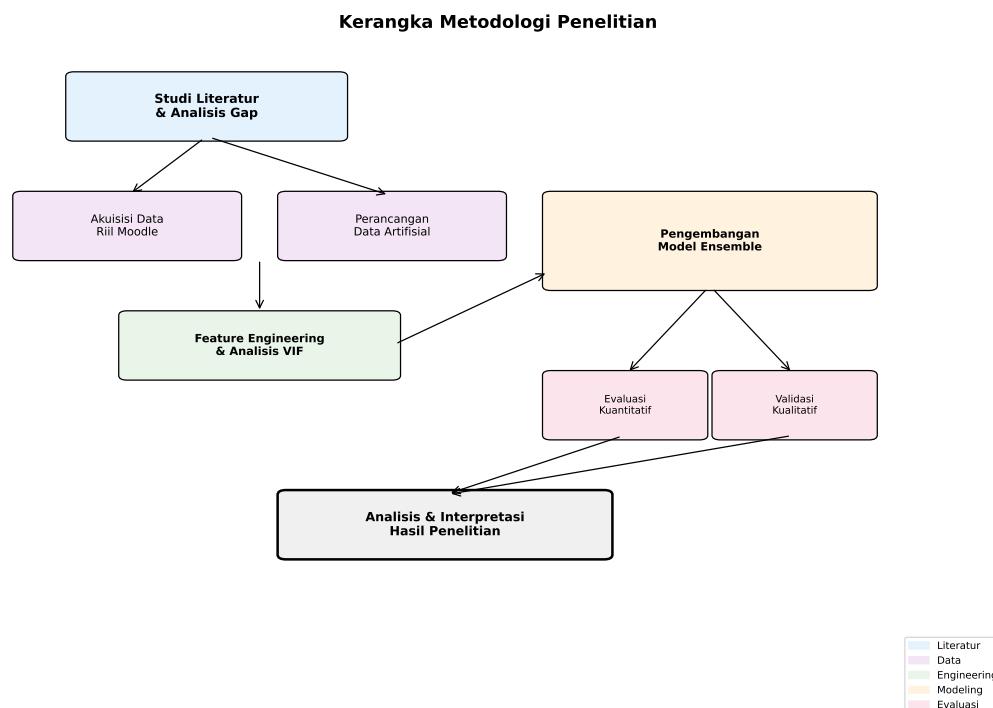
hanya andal secara teknis tetapi juga bertanggung jawab secara etis dan dapat diterapkan secara praktis menjadi fokus penting untuk penelitian selanjutnya.

Penelitian ini bertujuan untuk mengisi beberapa kesenjangan yang diidentifikasi dengan mengembangkan kerangka kerja komprehensif yang mengintegrasikan beberapa teknik deteksi sambil mempertimbangkan kendala praktis dan pertimbangan etis dalam konteks akademik.

BAB 3

METODE PENELITIAN

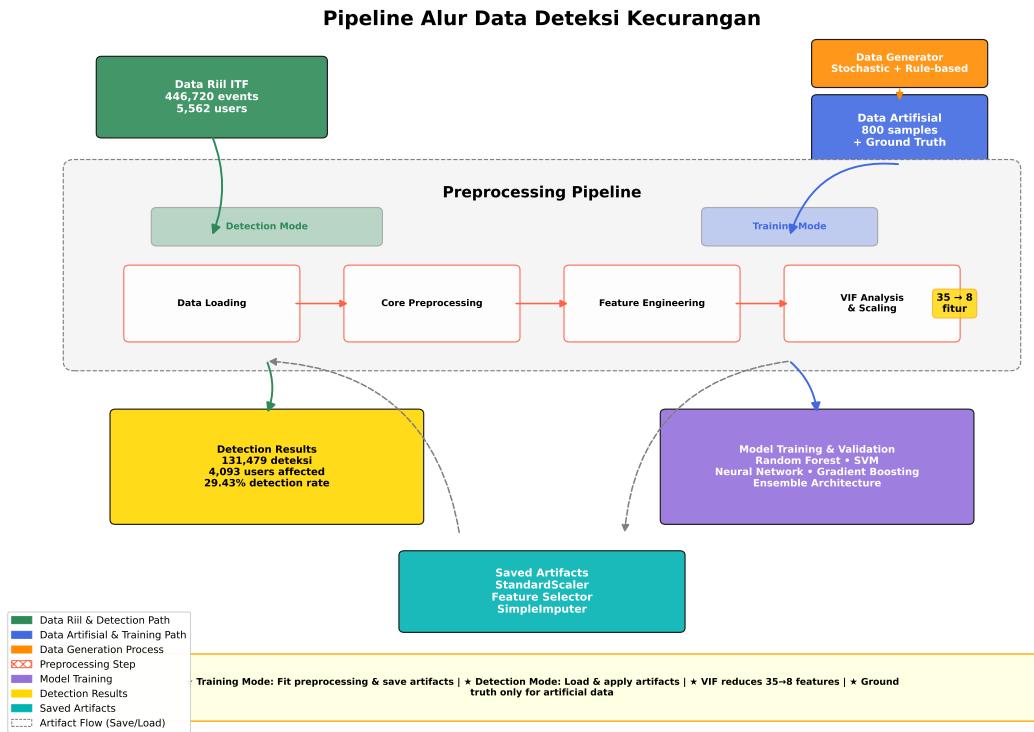
Bab ini memaparkan metode penelitian yang digunakan untuk mengembangkan sistem deteksi kecurangan akademik berbasis kecerdasan buatan pada platform Moodle. Metodologi penelitian dirancang melalui pendekatan sistematis yang dimulai dari pemahaman masalah, perancangan solusi, hingga implementasi dan evaluasi. Pembahasan mencakup enam fase utama: (1) analisis kebutuhan dan studi literatur, (2) akuisisi dan persiapan data, (3) perancangan data artifisial dengan *ground truth* terkontrol, (4) pengembangan *pipeline preprocessing* dan ekstraksi fitur, (5) pelatihan model *ensemble machine learning*, dan (6) evaluasi komprehensif pada data riil dan artifisial.



Gambar 3.1: Kerangka Metodologi Penelitian Deteksi Kecurangan

Gambar 3.1 menunjukkan kerangka metodologi penelitian yang terdiri dari enam fase utama yang saling berkaitan. Setiap fase dirancang secara spesifik untuk menjawab pertanyaan penelitian yang telah dirumuskan pada Bab 1. Fase 1 mengidentifikasi pola-pola kecurangan akademik melalui studi literatur. Fase 2 dan 3 mempersiapkan data pelatihan dengan *ground truth* yang terkontrol. Fase 4 mentransformasi data mentah menjadi fitur-fitur bermakna. Fase 5 mengembangkan model deteksi menggunakan pendekatan

ensemble. Fase 6 mengevaluasi efektivitas model pada skala operasional.



Gambar 3.2: Arsitektur Pipeline Teknis: Alur Pemrosesan Data dari Log Mentah hingga Deteksi Kecurangan dengan Dual-Mode Processing

Gambar 3.2 menerjemahkan kerangka metodologi konseptual menjadi implementasi teknis yang detail. *Pipeline* dimulai dari dua sumber data yang berbeda karakteristiknya: **Data Riil ITF** yang berisi 446.720 *events* dari aktivitas nyata mahasiswa Fasilkom UI, dan **Data Artifisial** dengan 800 sampel terkontrol yang dilengkapi *ground truth* untuk pelatihan model.

Perbedaan krusial terletak pada strategi pemrosesan: data artifisial diproses dalam *Training Mode* untuk membangun dan menyimpan *artifacts* pembelajaran (*scaler*, *feature selector*, *imputer*), sementara data riil menggunakan *Detection Mode* yang memanfaatkan *artifacts* tersebut untuk konsistensi transformasi. *Pipeline* berhasil mereduksi kompleksitas dari 35 fitur awal menjadi 8 fitur stabil melalui analisis *Variance Inflation Factor* (VIF), menghasilkan model *ensemble* yang mendeteksi 131.479 percobaan ujian mencurigakan (29,43% dari total).

Kombinasi kedua diagram memberikan pemahaman menyeluruh: Gambar 3.1 menyediakan peta konseptual untuk memandu pembaca melalui logika penelitian, sedangkan Gam-

bar 3.2 memberikan *blueprint* teknis untuk reproduksi penelitian. Integrasi pendekatan *top-down* (konseptual) dan *bottom-up* (teknis) ini memastikan transparansi metodologi yang dapat dipertanggungjawabkan secara ilmiah.

3.1 Desain Penelitian dan Hipotesis

Penelitian ini mengadopsi pendekatan eksperimental dengan kombinasi data artifisial dan riil untuk mengembangkan sistem deteksi kecurangan yang *robust* (kokoh) dan dapat diimplementasikan dalam skala institusional. Desain penelitian dirancang secara sistematis untuk menjawab pertanyaan penelitian utama: bagaimana mengembangkan sistem deteksi kecurangan otomatis yang akurat untuk platform Moodle menggunakan pendekatan *machine learning*?

3.1.1 Alur Tahapan Penelitian

Penelitian dilakukan melalui enam tahapan sistematis yang saling terkait:

1. **Fase 1 - Analisis Kebutuhan dan Studi Literatur:** Mengidentifikasi pola-pola kecurangan akademik dalam pembelajaran daring melalui *review* sistematis literatur. Fase ini menghasilkan pemahaman mendalam tentang karakteristik perilaku curang dan metode deteksi yang telah ada.
2. **Fase 2 - Akuisisi Data Log Moodle:** Mengumpulkan data log aktivitas dari platform Moodle Fasilkom UI dengan total 446.720 *events*. Data melalui proses anonimisasi untuk menjaga privasi pengguna.
3. **Fase 3 - Perancangan Data Artifisial:** Mengembangkan generator data artifisial yang mensimulasikan 800 skenario ujian dengan parameter kecurangan terkontrol. Setiap skenario dilengkapi *ground truth* untuk validasi model.
4. **Fase 4 - Pengembangan Pipeline Preprocessing:** Mengimplementasikan sistem ekstraksi dan transformasi fitur yang menghasilkan 35 fitur awal, kemudian direduksi menjadi 8 fitur stabil melalui analisis VIF.
5. **Fase 5 - Pelatihan Model Ensemble:** Melatih kombinasi algoritma *Random Forest*, *SVM*, *Neural Network*, dan *Gradient Boosting* dengan optimasi *hyperparameter* sistematis.
6. **Fase 6 - Evaluasi Multi-Aspek:** Menguji model pada data artifisial (evaluasi kuantitatif) dan mengaplikasikan pada data riil (evaluasi kualitatif) untuk memvalidasi

efektivitas deteksi.

Setiap fase dirancang dengan *output* yang jelas dan terukur, memungkinkan evaluasi progres penelitian secara objektif. Pendekatan iteratif diterapkan di mana hasil evaluasi dapat memicu perbaikan pada fase-fase sebelumnya.

3.1.2 Hipotesis Penelitian

Berdasarkan studi literatur dan analisis awal, penelitian ini menguji empat hipotesis utama:

H1: Deteksi Pola Kolaborasi

Kolaborasi kecurangan akan menghasilkan pola similaritas yang dapat dideteksi dalam tiga dimensi: navigasi (urutan penggerjaan soal), *timing* (pola waktu), dan jawaban (kesamaan respons). Hipotesis ini memprediksi tingkat akurasi deteksi $> 90\%$ untuk kasus dengan koordinasi tinggi.

H2: Superioritas Model Gabungan

Model *ensemble* (model gabungan) yang mengintegrasikan beberapa algoritma akan memberikan performa superior (peningkatan *F1-score* minimal 5%) dibandingkan model tunggal dalam mendekripsi berbagai strategi kecurangan yang heterogen.

H3: Threshold Ukuran Dataset

Dataset dengan minimal 500-1000 sampel diperlukan untuk mencapai performa deteksi yang optimal (akurasi $> 95\%$). Peningkatan ukuran dataset akan menghasilkan peningkatan performa yang signifikan hingga mencapai titik saturasi.

H4: Generalisasi Model

Model yang dilatih pada data artifisial dengan parameter terkontrol dapat menggeneralisasi dengan baik pada data riil skala besar, dengan tingkat deteksi yang konsisten dengan prevalensi kecurangan dalam literatur (20-40%).

Keempat hipotesis ini akan diuji secara empiris melalui eksperimen yang dirancang dalam metodologi penelitian, dengan hasil pengujian disajikan pada Bab 4.

3.2 Strategi Akuisisi dan Persiapan Data

Penelitian ini mengimplementasikan strategi *dual-data* yang menggabungkan kekuatan data riil (validitas eksternal) dan data artifisial (kontrol internal) untuk mengoptimalkan pengembangan model deteksi kecurangan. Strategi ini memungkinkan pelatihan model dengan *ground truth* yang terkontrol sambil mempertahankan kemampuan generalisasi pada kondisi operasional nyata.

3.3 Arsitektur Pipeline Preprocessing: Dari Log Mentah ke Fitur Terstruktur

Pipeline preprocessing data merupakan komponen krusial yang mentransformasi log mentah Moodle menjadi representasi fitur yang siap untuk *machine learning*. Sesuai dengan Fase 4 pada Gambar 3.1, *pipeline* ini memproses kedua jenis data (riil dan artifisial) melalui empat modul terintegrasi dengan strategi *dual-mode* yang memastikan konsistensi transformasi.

3.3.1 Komponen dan Alur Kerja Pipeline

Pipeline preprocessing dirancang dengan arsitektur modular yang terdiri dari empat komponen utama:

Modul I - Data Loading dan Validasi:

Modul pertama bertanggung jawab memuat data dari berbagai tabel Moodle dengan validasi komprehensif. Tabel-tabel utama yang diproses meliputi *mdl_quiz_attempts* (percobaan ujian), *mdl_question_attempt_steps* (langkah penggeraan), dan *mdl_question_attempt_step_data* (detail jawaban). Validasi mencakup pemeriksaan tipe data, konsistensi referensial antar tabel, dan penanganan nilai yang hilang atau *corrupt*. *Output* modul ini adalah dataset terintegrasi yang siap untuk *preprocessing* lanjutan.

Modul II - Core Preprocessing dan Normalisasi:

Modul kedua melakukan pembersihan dan normalisasi fundamental. Proses utama meliputi: (1) unifikasi *timestamp* ke format POSIX untuk konsistensi temporal, (2) penggabungan tabel-tabel terkait melalui *foreign key* untuk membentuk *event log* komprehensif, (3) *filtering* data berdasarkan kriteria penelitian seperti status *quiz completion*, dan (4) penghapusan duplikasi dan anomali data. Modul ini menghasilkan *clean dataset* dengan struktur yang konsisten.

Modul III - Feature Engineering Multi-Dimensi:

Modul ketiga mengekstraksi fitur-fitur *behavioral* dari *event log* yang telah dibersihkan. Ekstraksi fitur dilakukan dalam empat kategori:

- **Intra-Attempt Features:** Mengukur karakteristik dalam satu percobaan ujian seperti *total duration*, *number of actions*, dan *step duration statistics*
- **Sequential Features:** Menangkap pola urutan seperti *navigation patterns*, *revisit counts*, *linearity measures*, dan entropi navigasi
- **Similarity Features:** Menghitung kemiripan antar pengguna menggunakan *Levenshtein distance* untuk navigasi dan korelasi statistik untuk *timing patterns*
- **Comparative Features:** Menormalkan perilaku individual terhadap populasi menggunakan transformasi *z-score*

Modul IV - Feature Selection dan Optimization:

Modul terakhir melakukan optimasi fitur melalui beberapa tahap:

- **Missing Value Imputation:** Menggunakan *SimpleImputer* dengan strategi *mean* untuk fitur numerik
- **Multicollinearity Analysis:** Menghitung *Variance Inflation Factor* (VIF) dan mengeliminasi fitur dengan $VIF > 10$
- **Variance Filtering:** Menghilangkan fitur dengan $variance < 0,01$ yang tidak informatif
- **Standardization:** Menerapkan *StandardScaler* untuk normalisasi distribusi fitur

3.3.2 Strategi Dual-Mode Processing

Inovasi kunci dalam *pipeline* adalah implementasi *dual-mode processing* yang membedakan pemrosesan data *training* dan *detection*:

Training Mode untuk Data Artifisial:

Dalam *mode training*, *pipeline* melakukan *fitting* terhadap data artifisial dan menyimpan semua *transformation artifacts* ke direktori *preprocessing/artifacts/*. *Artifacts* yang disimpan meliputi: (1) *fitted imputer* untuk menangani *missing value*, (2) *fitted scaler* dengan parameter *mean* dan *standard deviation*, (3) *feature selector* dengan daftar fitur terpilih, dan (4) *metadata* transformasi untuk *reproducibility*. Mode ini memungkinkan eksplorasi parameter *preprocessing* optimal melalui eksperimen terkontrol.

Detection Mode untuk Data Riil:

Dalam *mode detection*, *pipeline* memuat *artifacts* yang telah disimpan dan menerapkannya pada data riil tanpa melakukan *fitting* ulang. Pendekatan ini memastikan: (1) konsistensi transformasi antara *training* dan *detection*, (2) pencegahan *data leakage* dari *test set*, (3) efisiensi komputasi dengan menghindari *re-fitting*, dan (4) *reproducibility* hasil deteksi. *Pipeline* secara otomatis mendeteksi keberadaan *artifacts* dan beralih ke mode yang sesuai.

Strategi *dual-mode* ini merupakan *best practice* dalam *machine learning* yang memastikan validitas metodologis sambil mempertahankan efisiensi operasional. Transparansi penuh dalam proses transformasi memungkinkan audit dan verifikasi ilmiah terhadap setiap tahap *preprocessing*.

Data Riil Moodle:

Data log *Moodle* diperoleh langsung dari sistem yang dikelola oleh tim ITF Fasilkom UI dan telah melalui proses anonimisasi untuk menjaga privasi pengguna. Data ini digunakan untuk validasi model dalam konteks operasional nyata.

Data Artifisial:

Data artifisial dirancang khusus untuk pelatihan model dengan *ground truth* yang terdokumentasi. Pendekatan ini memungkinkan eksplorasi berbagai skenario kecurangan dan kontrol parameter yang tidak mungkin dilakukan pada data riil.

3.3.3 Data Log Moodle Riil: Deskripsi (periode, jumlah event/user, fitur utama), Proses Akuisisi, Kebijakan Anonimisasi & Etika.

Subbab ini menjelaskan mengenai data log yang diambil langsung dari sistem *Moodle*, yang dikelola oleh tim ITF Fasilkom UI dan disimpan pada Lumbung Storage Cloud (mirip dengan platform penyimpanan seperti Google Drive). Data yang digunakan telah melalui proses anonimisasi, di mana identitas asli pengguna (username atau nama lengkap) tidak disertakan, melainkan hanya diwakili oleh *user_id*.

3.3.3.1 Deskripsi Data

Data log *Moodle* terdiri dari beberapa tabel utama yang masing-masing menyimpan informasi berbeda terkait aktivitas pengguna dan kuis. Berikut adalah rincian kolom-kolom yang terdapat dalam tiap tabel:

mdl_question_usages

Kolom: *question_usage_id, context_id*

Menyimpan informasi terkait konteks penggunaan pertanyaan dalam kuis.

mdl_quiz_grades

Kolom: *quiz_grades_id, quiz_id, user_id, final_grade*

Berisi nilai akhir dari masing-masing kuis yang diambil oleh pengguna.

mdl_question_attempt_steps

Kolom: *question_step_id, question_attempt_id, sequencenumber, state, timecreated*

Mencatat tiap langkah (*step*) dalam upaya pengerjaan soal, termasuk status dan waktu pembuatan.

mdl_quiz_attempts

Kolom: *attempt_id, quiz_id, user_id, question_usage_id, timestart, timefinish, state, sum-grades*

Menggambarkan detail setiap upaya pengerjaan kuis, termasuk waktu mulai, waktu selesai, status, dan jumlah nilai yang diperoleh.

Catatan: Terdapat anomali pada *timefinish* (misalnya *timestamp* 1970-01-01) yang kemungkinan menunjukkan upaya kuis yang belum selesai atau nilai *default* dari *Unix epoch*.

mdl_question_answers

Kolom: *question_answers_id, questionid, answer_text, fraction*

Menyimpan data mengenai jawaban yang diberikan pada tiap pertanyaan, termasuk teks jawaban dan bobot nilai yang terkait.

mdl_question_attempt_step_data

Kolom: *step_data_id, question_step_id, name, value*

Menyimpan data tambahan terkait langkah pengerjaan soal, yang dapat berupa nilai-nilai pendukung dari proses evaluasi.

mdl_quiz

Kolom: *quiz_id, course, quiz_name, timeopen, timeclose, timelimit*

Berisi informasi dasar mengenai kuis, termasuk mata kuliah, nama kuis, serta waktu buka dan tutup kuis.

mdl_sessions

Kolom: *session_id, user_id, timecreated, lastip, sessdata*

Mencatat aktivitas sesi pengguna, mulai dari waktu pembuatan sesi hingga informasi terkait IP dan data sesi lainnya.

3.3.3.2 Rentang Waktu dan Skala Dataset

Data log *Moodle* mencakup periode hampir 10 tahun, dengan rentang data keseluruhan dari tanggal 31 Juli 2015 hingga 22 Februari 2025. Secara spesifik:

mdl_question_attempt_steps:

Rentang waktu: 29 Agustus 2015–22 Februari 2025

Durasi: sekitar 3.565 hari

mdl_quiz_attempts:

Rentang waktu untuk *timestart*: 31 Juli 2015–22 Februari 2025

Rentang waktu untuk *timefinish*: (dengan catatan anomali *timestamp* 1970 sebagai *default*) hingga 22 Februari 2025

Durasi: sekitar 3.594 hari (mengabaikan nilai *default* 1970)

mdl_sessions:

Rentang waktu: 17 Maret 2020–22 Februari 2025

Durasi: sekitar 1.803 hari

Skala data secara keseluruhan meliputi:

- Total kuis yang diambil: 446.720 upaya
- Jumlah pengguna unik: 5.562
- Jumlah kuis unik: 6.304
- Jumlah langkah pertanyaan: 22.192.809

3.3.3.3 Cakupan Mata Kuliah dan Pola Penggunaan

Data ini mencakup aktivitas di lebih dari 140 Mata Kuliah unik, dengan variasi ukuran kelas:

Ukuran Mata Kuliah:

- Mata Kuliah besar (300+ mahasiswa): sekitar 10 Mata Kuliah
- Mata Kuliah menengah (100-300 mahasiswa): sekitar 30 Mata Kuliah
- Mata Kuliah kecil (< 100 mahasiswa): sekitar 100 Mata Kuliah
- Mata Kuliah sangat kecil (< 10 mahasiswa): sekitar 15 Mata Kuliah

Contoh Mata Kuliah dengan aktivitas tinggi:

- Mata Kuliah 3836: 453 peserta, 5.544 upaya (rata-rata 12,24 upaya per pengguna)
- Mata Kuliah 3634: 442 peserta, 5.902 upaya (rata-rata 13,35 upaya per pengguna)
- Mata Kuliah 3723: 390 peserta, 3.829 upaya (rata-rata 9,56 upaya per pengguna)
- Mata Kuliah 3640: 386 peserta, 4.463 upaya (rata-rata 11,56 upaya per pengguna)
- Mata Kuliah 3636: 382 peserta, 761 upaya (rata-rata 1,99 upaya per pengguna)

Pola penggunaan kuis menunjukkan perbedaan yang signifikan antar Mata Kuliah. Beberapa Mata Kuliah hanya mencatat satu upaya per pengguna, sedangkan Mata Kuliah lain mencatat rata-rata 10 upaya atau lebih, yang mengindikasikan adanya kuis latihan atau kebijakan pengulangan untuk meningkatkan pemahaman materi.

3.3.3.4 Proses Akuisisi dan Kebijakan Anonimisasi

Data log *Moodle* diakuisisi secara langsung dari sistem *Moodle* oleh tim ITF Fasilkom UI dan kemudian disimpan di Lumbung Storage Cloud. Proses akuisisi melibatkan:

Pengambilan Data:

Data diekstraksi dari server *Moodle* dengan menggunakan prosedur logging yang telah ditetapkan, memastikan setiap aktivitas terekam secara lengkap.

Pengiriman dan Penyimpanan:

Data dikirim dan disimpan secara terpusat di Lumbung Storage Cloud yang merupakan *repository* internal Fasilkom UI, menjamin keamanan dan integritas data.

Anonimisasi:

Untuk melindungi privasi pengguna, data telah diproses sehingga informasi identitas pribadi (username, nama lengkap, dan data sensitif lainnya) dihilangkan. Hanya *user_id* yang tetap dipertahankan untuk keperluan analisis. Hal ini sesuai dengan standar etika penelitian dan kebijakan perlindungan data yang berlaku.

Pertimbangan Etika:

Penggunaan data riil ini telah mempertimbangkan aspek etika dan regulasi perlindungan data. Kebijakan anonimisasi yang diterapkan memastikan bahwa data tidak dapat dikaitkan langsung dengan individu tertentu, sehingga menjaga kerahasiaan dan privasi pengguna.

3.3.4 Strategi Data Artifisial

Data artifisial dalam penelitian ini dirancang untuk mengatasi keterbatasan *ground truth* pada data riil. Strategi ini memungkinkan kontrol penuh terhadap parameter kecurangan dan validasi objektif terhadap performa model, dengan alasan-alasan sebagai berikut:

1. Kontrol Variabel dan Simulasi Skenario Ekstrem:

Data artifisial memungkinkan penciptaan skenario perilaku pengguna yang ekstrem atau tidak biasa, yang mungkin jarang terjadi pada data log riil. Dengan demikian, model dapat dilatih untuk mengenali pola-pola *non-compliance* secara lebih spesifik dan mendalam.

2. Pengembangan Model yang Terarah:

Dalam penelitian ini, pengembangan model dilakukan sepenuhnya dengan menggunakan data artifisial. Pendekatan ini memberikan fleksibilitas dalam mengatur parameter dan iterasi pelatihan, serta mengurangi risiko *overfitting* pada data riil yang belum terlabel dengan jelas.

3. Evaluasi Model dengan Data Riil:

Setelah model dioptimasi dengan data artifisial, evaluasi akhir akan dilakukan dengan menerapkan model pada data log asli. Langkah ini bertujuan untuk mengukur sejauh mana model dapat mendeteksi kasus kecurangan yang benar-benar terjadi di lingkungan sistem *Moodle*, sehingga memberikan validasi nyata terhadap efektivitas pendekatan yang digunakan.

4. Pendekatan Metodologis dalam Pembuatan Data Artifisial:

Data artifisial dihasilkan melalui kombinasi simulasi berbasis aturan dan proses stokastik, yang disesuaikan dengan pola penggunaan yang ditemukan dalam data log riil. Pendekatan ini memastikan bahwa data artifisial tidak hanya mereplikasi kondisi normal, tetapi juga memasukkan skenario *non-compliance* yang relevan untuk pengembangan model.

Dengan demikian, penggunaan data artifisial dalam penelitian ini memberikan keuntungan

dalam hal kontrol variabilitas dan eksplorasi skenario ekstrem, serta mempercepat proses pengembangan model. Evaluasi akhir dengan data log riil akan menjadi tolok ukur untuk menilai kemampuan model dalam mendeteksi kecurangan yang terjadi secara nyata dalam sistem *Moodle*.

3.4 Preprocessing Pipeline dan Feature Engineering

Tahap *feature engineering* dan analisis VIF merupakan komponen kritis dalam *pipeline* penelitian ini. Proses ini mencakup transformasi data mentah menjadi representasi fitur yang bermakna, analisis multikolinearitas, dan seleksi fitur optimal untuk model *machine learning*.

Gambar 3.3 menunjukkan alur transformasi data dari *raw logs* hingga *detection output*. Reduksi dramatis dari 446.720 *events* menjadi 800 *training samples* (faktor 560x) dilakukan melalui agregasi per *user-quiz attempt*. *Feature extraction* menghasilkan 35 fitur awal yang kemudian direduksi menjadi 8 fitur stabil melalui *VIF analysis*. Data dibagi dengan proporsi 70/15/15 untuk *training*, *validation*, dan *test* untuk memastikan evaluasi yang adil dan mencegah *overfitting*.

Proses pra-pemrosesan tidak hanya bertujuan untuk mengurangi gangguan dan inkonsistensi, tetapi juga untuk memastikan bahwa fitur-fitur yang dihasilkan mencerminkan karakteristik perilaku pengguna secara akurat. Langkah-langkah berikut diambil dengan dasar metodologis yang dapat dipertahankan secara ilmiah:

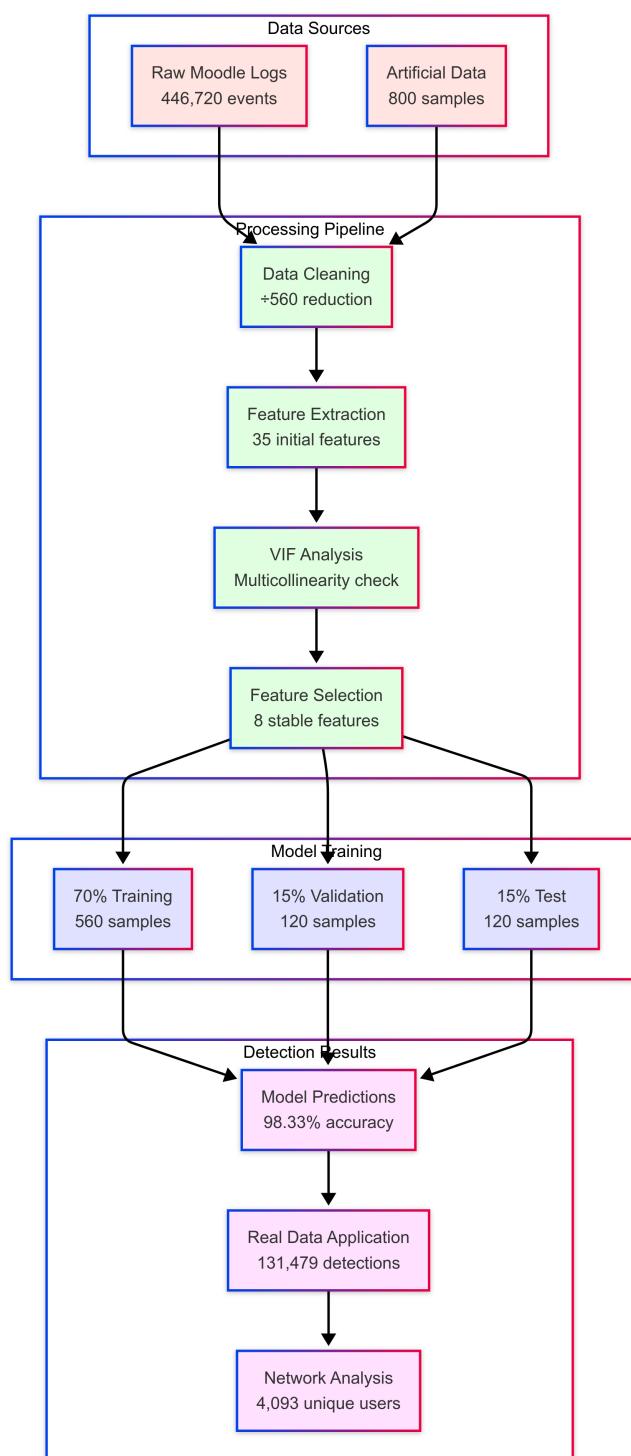
3.4.1 Pembersihan Data (Data Cleaning)

Penanganan Missing Values:

Data log sering kali mengandung nilai yang hilang (*missing values*) karena ketidakteraturan dalam pencatatan *event* atau *error* saat pengambilan data. Dalam *pipeline*, nilai yang tidak terisi diimputasi dengan metode penggantian menggunakan nilai *default* (misalnya, 0) atau perhitungan statistik (rata-rata/median) ketika relevan. Misalnya, dalam skrip `preprocess_features.py`, setelah ekstraksi fitur, seluruh nilai NaN diisi dengan nol untuk memastikan tidak ada celah data yang dapat mengganggu analisis model.

Filtering Data yang Tidak Relevan:

Dalam konteks deteksi kecurangan, tidak seluruh *event log* memiliki nilai informasi yang



Gambar 3.3: Alur Data dalam *Pipeline* Deteksi Kecurangan

sama. Oleh karena itu, dilakukan penyaringan untuk:

- Mengabaikan *event* dengan atribut tertentu (misalnya, *field contextlevel* yang bernilai '*system*') karena *event* ini tidak mewakili aktivitas pengguna pada kuis.
- Menghapus entri dengan *user_id* yang *null*, mengingat identifikasi pengguna merupakan variabel kunci dalam analisis pola perilaku.

3.4.2 Transformasi dan Normalisasi Data

Unifikasi Format Waktu dan Normalisasi Zona Waktu:

Data log mengandung *timestamp* yang berasal dari sumber atau zona waktu yang berbeda. Untuk memastikan keseragaman, seluruh *timestamp* dikonversi ke dalam format numerik, seperti *Unix timestamp* (atau ISO 8601 jika diperlukan), melalui fungsi konversi di `preprocess_features.py`. Proses ini juga melibatkan normalisasi zona waktu agar seluruh *event log* dapat dibandingkan secara akurat dalam kerangka waktu yang sama.

Parsing Nested Fields:

Banyak kolom dalam data log disimpan dalam bentuk *string* yang merepresentasikan *array* atau *dictionary* (misalnya, *sequence* navigasi atau *transition times*). Menggunakan fungsi seperti `ast.literal_eval`, skrip melakukan *parsing* terhadap *string* tersebut sehingga struktur data yang tersusun (*list* atau *dictionary*) dapat diekstraksi. Proses ini memungkinkan perhitungan fitur statistik seperti panjang *sequence*, entropi, dan jumlah *revisits*, yang esensial untuk mendeteksi pola aktivitas pengguna.

3.4.3 Ekstraksi Fitur dan Deteksi Outlier

Setelah data dibersihkan dan ditransformasikan, tahap selanjutnya adalah ekstraksi fitur, di mana fitur-fitur dasar dan lanjutan dihasilkan untuk mendukung proses pelatihan model deteksi kecurangan. Dalam *pipeline*, ekstraksi fitur dilakukan melalui modul `feature_eng.py` dan `fixed_extraction.py`, dengan beberapa langkah sebagai berikut:

Fitur Dasar:

Fitur seperti jumlah percobaan kuis, rata-rata waktu penggeraan, total waktu, serta statistik minimum dan maksimum dihitung dari data log kuis. Contoh implementasi terdapat pada fungsi `extract_basic_statistics()` di `feature_eng.py`, yang juga menyertakan agregasi data per pasangan *user-kuis*.

Fitur Sequence:

Dari data navigasi dan jawaban, diekstraksi fitur-fitur seperti:

- Panjang *sequence* dan jumlah pertanyaan unik: Mengukur seberapa panjang dan bervariasinya aktivitas navigasi pengguna.
- *Linearity*: Menghitung seberapa berurutan langkah-langkah yang diambil, menggunakan perhitungan rasio antara pertanyaan unik dan total langkah.
- *Revisits*: Penghitungan jumlah *revisits* atau langkah yang diulang, sebagai indikasi pola abnormal.

Fungsi `extract_navigation_features()` dalam `fixed_extraction.py` mengilustrasikan bagaimana fitur-fitur tersebut dihitung dengan memanfaatkan evaluasi *list* dari *sequence*.

Deteksi Outlier pada Fitur Waktu:

Pada ekstraksi fitur waktu, dilakukan perhitungan statistik seperti rata-rata, standar deviasi, nilai minimum dan maksimum dari durasi pengerjaan soal. Selain itu, skrip juga menghitung jumlah *event* dengan durasi sangat pendek (misalnya, < 5 detik) dan sangat panjang (misalnya, > 600 detik).

Berikut adalah cuplikan kode dari fungsi `extract_timing_features()` yang digunakan untuk mendeteksi pola waktu yang abnormal, yang dapat dijadikan indikator *outlier*:

```
def extract_timing_features(time_seq):
    """Ekstraksi fitur dari sequence waktu."""
    features = {}

    try:
        if isinstance(time_seq, str):
            time_seq = eval(time_seq)
        time_seq = np.array(time_seq, dtype=float)

        features['mean_time'] = float(np.mean(time_seq))
        features['std_time'] = float(np.std(time_seq))
        features['min_time'] = float(np.min(time_seq))
```

```

        features['max_time'] = float(np.max(time_seq))

        # Pola waktu mencurigakan sebagai indikator outlier
        features['quick_answers'] = int(sum(1 for t in time_seq if t < 5))
        features['very_long_answers'] = int(sum(1 for t in time_seq if t > 600))
    except Exception as e:
        print(f"Warning: Error processing timing features: {e}")
        features['mean_time'] = 0.0
        features['std_time'] = 0.0
        features['min_time'] = 0.0
        features['max_time'] = 0.0
        features['quick_answers'] = 0
        features['very_long_answers'] = 0

    return features

```

Nilai `quick_answers` dan `very_long_answers` ini memberikan gambaran tentang berapa kali pengguna menyelesaikan bagian tertentu dengan durasi yang sangat tidak wajar, yang kemudian dapat diintegrasikan sebagai fitur untuk mendeteksi perilaku kecurangan.

Penghitungan *Similarity Features*:

Untuk mendeteksi kolaborasi kecurangan, *pipeline* menghitung matriks kemiripan (*similarity matrix*) berdasarkan pola navigasi, waktu, dan jawaban antar pengguna. Fungsi `calculate_similarity_matrices()` di `feature_eng.py` menghitung kemiripan menggunakan metode seperti *Levenshtein distance* untuk *sequence* navigasi dan korelasi untuk *sequence* waktu, kemudian hasilnya digunakan untuk menambahkan fitur agregat seperti rata-rata dan maksimum *similarity* antar pengguna.

3.4.4 Checklist Pra-pemrosesan

Untuk memastikan bahwa seluruh proses pra-pemrosesan dapat diulangi dan diverifikasi, disusunlah *checklist* sebagai berikut:

- Konversi seluruh *timestamp* ke format standar (*Unix timestamp* atau ISO 8601).
- Penghapusan *event* dengan atribut *contextlevel* bernilai '*system*'.

- Penyaringan entri dengan *user_id null*.
- *Parsing* kolom yang berisi *string* representasi *array/dict* untuk ekstraksi fitur.
- Normalisasi zona waktu untuk keseragaman data.
- Perhitungan statistik fitur dasar dan *sequence* (termasuk deteksi pola *outlier* pada waktu).

3.4.5 Justifikasi Ilmiah

Pendekatan pra-pemrosesan yang diterapkan dalam penelitian ini didasarkan pada prinsip-prinsip *data cleaning* dan *feature engineering* yang telah terbukti secara empiris meningkatkan kualitas *dataset* dan kinerja model *machine learning*.

Pembersihan dan Transformasi:

Dengan memastikan bahwa data dalam format yang konsisten dan bebas dari *missing values*, variabilitas yang tidak relevan dapat diminimalkan, sehingga model tidak terdistorsi oleh gangguan data.

Ekstraksi Fitur:

Fitur-fitur yang diekstraksi, seperti *linearity* dan *revisits* pada *sequence*, memberikan representasi numerik yang dapat menggambarkan perilaku pengguna secara mendalam.

Deteksi Outlier:

Dengan mengidentifikasi *event* dengan durasi yang sangat singkat atau sangat panjang, *pipeline* mampu memberikan sinyal peringatan terhadap kemungkinan aktivitas yang tidak wajar, yang secara langsung berkontribusi pada identifikasi kecurangan.

Reproducibility:

Checklist dan struktur modular pada skrip memastikan bahwa seluruh proses dapat direplikasi dan diaudit, sehingga mendukung validitas dan *reproducibility* dari penelitian.

Melalui serangkaian proses yang sistematis dan berbasis algoritma yang telah teruji, tahap pra-pemrosesan ini memberikan dasar yang kuat untuk langkah-langkah selanjutnya dalam *pipeline*, yaitu ekstraksi fitur lanjutan, pelatihan model, dan akhirnya evaluasi deteksi kecurangan.

3.4.6 Perancangan dan Generasi Data Artifisial

Subbab ini menjelaskan secara mendalam mengenai rancangan, implementasi, dan validasi data log *Moodle* artifisial yang digunakan untuk mengembangkan model deteksi *non-compliance*. Pendekatan yang diterapkan dikenal dengan istilah *Skenario Perilaku Sintetik*, yaitu simulasi aktivitas pengguna (baik perilaku normal maupun *non-compliance*) melalui algoritma yang menggabungkan simulasi berbasis aturan dan proses stokastik. Data artifisial ini tidak hanya mereplikasi aktivitas log riil, tetapi juga memungkinkan eksplorasi skenario ekstrem yang jarang terekam pada data nyata, seperti koordinasi kelompok kecurangan dengan pola sinkronisasi tinggi.

3.4.7 Definisi Operasional Skenario Perilaku Sintetik

Skenario Perilaku Sintetik didefinisikan sebagai rangkaian aturan dan mekanisme yang dirancang untuk mereplikasi pola aktivitas pengguna pada sistem *Moodle*, baik dalam kondisi normal maupun *non-compliance* (kecurangan). Pendekatan ini mengintegrasikan simulasi berbasis aturan dengan proses stokastik, sehingga memungkinkan pembentukan data log artifisial yang tidak hanya mereplikasi aktivitas log riil, tetapi juga dapat mengeksplorasi skenario ekstrem yang jarang terekam dalam data nyata.

Perilaku Normal:

Pada skenario perilaku normal, aktivitas pengguna dirancang untuk mencerminkan dinamika berpikir yang alami, yang ditandai dengan:

- Variasi Urutan Navigasi: Pengguna normal menunjukkan urutan akses pertanyaan yang bervariasi, dengan kemungkinan revisi jawaban yang berbeda-beda antar sesi. Hal ini menggambarkan proses evaluasi internal yang dinamis.
- Variasi Pola Jawaban: Pola jawaban mencerminkan respon acak yang wajar, dengan proporsi jawaban benar dan salah yang bervariasi secara alami.
- Waktu Pengerjaan yang Variatif: Interval waktu antar pertanyaan bervariasi, mencerminkan kecepatan berpikir dan penyesuaian terhadap tingkat kesulitan pertanyaan. Distribusi waktu pengerjaan ini umumnya menunjukkan standar deviasi yang wajar.

Perilaku *Non-Compliance* (Kecurangan):

Pada skenario *non-compliance*, pola aktivitas sengaja diatur untuk menciptakan indikasi kecurangan, dengan karakteristik sebagai berikut:

- Kesamaan Navigasi: Urutan pertanyaan yang diakses oleh pengguna dalam kelompok kecurangan sangat mirip, termasuk adanya revisi yang identik antar anggota kelompok.
- Konsistensi Jawaban: Pola jawaban menunjukkan tingkat keseragaman yang tinggi. Misalnya, terdapat kecenderungan anggota kelompok secara bersama-sama memberikan jawaban salah pada pertanyaan-pertanyaan sulit.
- Sinkronisasi Waktu yang Tidak Wajar: Interval waktu antar pertanyaan hampir seragam antar pengguna. Dalam beberapa kasus, terdapat kelompok yang menyelesaikan kuis secara bersamaan dalam waktu kurang dari 15 detik, dengan standar deviasi yang rendah, mengindikasikan adanya pola sinkronisasi yang sulit terjadi secara alami.
- Mekanisme *Leader-Follower*: Terdapat pola di mana satu anggota (*leader*) menyelesaikan kuis terlebih dahulu, sedangkan anggota lainnya (*follower*) mengikuti dengan *delay* yang konsisten. Pola ini menciptakan korelasi tinggi dalam waktu pengerjaan antar anggota, yang menjadi indikator kuat adanya koordinasi.

Definisi operasional ini menjadi dasar untuk memformulasikan parameter simulasi. Parameter-parameter tersebut, seperti tingkat kemiripan navigasi, *delay* waktu, dan distribusi jawaban, kemudian diintegrasikan dalam algoritma generasi data log artifisial. Setiap skenario diberikan label *ground truth*, yang memungkinkan validasi dan evaluasi model deteksi kecurangan secara kuantitatif dan kualitatif.

3.4.8 Desain *Ground Truth* Artifisial

Dalam proses generasi data log artifisial, setiap entitas—baik pada level sesi, upaya pengerjaan, maupun langkah pertanyaan—diberi label *ground truth* yang mendefinisikan status sebagai aktivitas normal atau *non-compliance*. Desain *ground truth* ini tidak hanya merupakan hasil dari parameter simulasi, tetapi juga dilengkapi dengan dokumentasi yang komprehensif melalui file `cheating_ground_truth.md`. File ini berfungsi sebagai acuan empiris untuk validasi dan evaluasi model deteksi kecurangan.

Komponen Utama *Ground Truth*:

- **Label Kecurangan:**

Setiap entitas dilengkapi dengan label:

- 0: Menunjukkan aktivitas normal.
- 1: Menunjukkan aktivitas *non-compliance* (kecurangan), yang dihasilkan dari simu-

lasi kelompok dengan pola sinkronisasi tinggi dan koordinasi yang jelas.

- **Komposisi Dataset:**

Data artifisial dihasilkan dengan proporsi tertentu antara perilaku normal dan *non-compliance*. Proporsi ini dikonfigurasi dalam parameter simulasi, misalnya 10–20% pengguna disimulasikan sebagai *non-compliance*, untuk memastikan keseimbangan yang memadai dalam pelatihan dan evaluasi model.

- **Pencatatan Parameter Simulasi dan Contoh Kasus:**

Parameter-parameter yang mendasari penetapan label *ground truth* direkam secara detail, mencakup:

- *Navigation Similarity*: Persentase kesamaan urutan navigasi antar anggota kelompok (contoh: 96%).
- *Answer Pattern Similarity*: Persentase kesamaan pola jawaban (contoh: 94%).
- *Timing Correlation*: Koefisien korelasi waktu antar pertanyaan (contoh: 0,95).
- *Standard Deviation* (Rata-rata): Rata-rata standar deviasi waktu pengerjaan per pertanyaan (contoh: 12 detik).
- *Wrong Answer Bias*: Probabilitas kesalahan terkoordinasi (contoh: 87%).

Nilai-nilai di atas disajikan sebagai contoh dalam file `cheating_ground_truth.md`. Penting untuk dicatat bahwa angka-angka tersebut merupakan representasi sampel yang dapat dikonfigurasi ulang sesuai kebutuhan eksperimen, dan bukan merupakan nilai final yang harus diterapkan secara universal.

Dokumentasi Melalui File Cheating Ground Truth:

File `cheating_ground_truth.md` menyajikan tabel ringkasan statistik kelompok kecurangan beserta parameter-parameter yang telah diukur secara simulatif. Contoh tabel tersebut mencakup:

- Kelompok Kecurangan dengan *Severity* Tinggi: Menunjukkan kesamaan navigasi, pola jawaban, dan korelasi waktu yang sangat tinggi, serta standar deviasi dan *wrong answer bias* yang rendah.
- Kelompok Kecurangan dengan *Severity* Sedang: Menunjukkan nilai yang lebih moderat, dengan *delay* waktu dan variansi yang lebih besar.

Dokumentasi ini menyediakan bukti empiris dan justifikasi statistik bahwa skenario *non-compliance* yang disimulasikan memiliki karakteristik yang berbeda secara signifikan dari

aktivitas normal. Informasi ini sangat penting untuk validasi model, karena memungkinkan perbandingan langsung antara prediksi model dengan *ground truth*.

Integrasi dan Reproducibility:

Ground truth artifisial yang terdokumentasi secara rinci ini diintegrasikan langsung ke dalam dataset log artifisial. Pendekatan ini memastikan bahwa evaluasi model dapat dilakukan secara objektif menggunakan metrik seperti *precision*, *recall*, *F1-score*, dan akurasi. Selain itu, pencatatan parameter simulasi dan contoh kasus dalam file `cheating_ground_truth.md` mendukung *reproducibility* penelitian, karena eksperimen dapat diulang dengan menggunakan *seed* dan konfigurasi yang sama.

Dengan demikian, desain *ground truth* artifisial ini tidak hanya menyediakan label yang diperlukan untuk evaluasi model, tetapi juga memberikan kerangka kerja untuk analisis sensitivitas dan validasi statistik, yang secara bersama-sama mendukung pendekatan *Skenario Perilaku Sintetik* yang digunakan dalam penelitian ini.

3.4.9 Metode Generasi Data

Proses generasi data log artifisial dilakukan dengan pendekatan yang menggabungkan simulasi berbasis aturan dan proses stokastik, sehingga dapat mereplikasi pola aktivitas pengguna pada sistem *Moodle* secara realistik. Pendekatan ini juga memungkinkan eksplorasi skenario ekstrem *non-compliance* yang jarang terekam pada data riil. Metode yang diterapkan meliputi beberapa komponen kunci berikut:

Simulasi Berbasis Aturan:

Algoritma simulasi menetapkan seperangkat aturan untuk menentukan urutan navigasi, pola revisi jawaban, dan distribusi waktu antar pertanyaan. Aturan-aturan ini dirancang untuk meniru dinamika aktivitas pengguna normal maupun perilaku kecurangan. Sebagai contoh, pada kelompok *non-compliance*, aturan mensyaratkan agar setiap anggota mengikuti urutan pertanyaan yang seragam dan menunjukkan kesamaan dalam revisi jawaban, sehingga menghasilkan tingkat kesamaan navigasi dan pola jawaban yang tinggi.

Proses Stokastik:

Untuk menghindari keteraturan yang sempurna dan menambahkan tingkat realisme, proses stokastik diterapkan dalam penentuan waktu pengerjaan, revisi jawaban, dan urutan pertanyaan. Penggunaan fungsi `random` dengan *seed* yang telah ditentukan memungkinkan

variasi terkontrol, sehingga setiap simulasi menghasilkan distribusi waktu dan pola yang mendekati kondisi riil, namun masih mempertahankan karakteristik khas dari skenario *non-compliance*.

Modifikasi Berdasarkan Data Riil:

Parameter dasar seperti `base_date`, `timelimit`, serta distribusi waktu penggeraan diadaptasi dari analisis data log *Moodle* riil. Pendekatan ini memastikan bahwa data artifisial tidak hanya bersifat sintetik, tetapi juga memiliki kemiripan statistik dengan data riil. Dengan demikian, model yang dikembangkan dapat diuji dan divalidasi secara lebih representatif terhadap kondisi operasional di lingkungan nyata.

Penerapan Mekanisme *Leader-Follower*:

Untuk mengantisipasi skenario kecurangan dengan tingkat koordinasi yang tinggi, algoritma juga mengimplementasikan pola *leader-follower*. Dalam mekanisme ini, satu anggota (*leader*) menyelesaikan kuis terlebih dahulu dengan pola waktu yang stabil, sementara anggota lain (*follower*) mengikuti dengan *delay* konsisten yang telah disimulasi. Penghitungan korelasi waktu antar anggota yang tinggi (misalnya, nilai $> 0,8$) menjadi indikator kuat atas adanya koordinasi, sehingga pola ini diintegrasikan dalam parameter simulasi.

Parameter Simulasi dan Konfigurasi:

Seluruh proses generasi data dikonfigurasi melalui *file* konfigurasi (misalnya, *file JSON*) yang menetapkan parameter-parameter penting, seperti:

- Jumlah total pengguna dan kuis.
- Proporsi pengguna dengan perilaku *non-compliance*.
- Batasan waktu (`timelimit`) dan `base_date` sebagai dasar simulasi.
- Nilai *threshold* untuk *similarity* dalam pola navigasi, jawaban, dan waktu.

Parameter-parameter ini bersifat fleksibel dan dapat disesuaikan untuk mengeksplorasi berbagai skenario, mulai dari aktivitas normal hingga skenario ekstrem yang memicu koordinasi *non-compliance* secara signifikan.

Melalui kombinasi simulasi berbasis aturan dan proses stokastik, metode generasi data ini mampu menghasilkan log aktivitas yang kaya informasi dan mendekati realitas, sekali-gus memungkinkan pengujian model dalam berbagai kondisi ekstrem. Pendekatan ini

memberikan dasar yang solid untuk evaluasi dan validasi model deteksi *non-compliance*, dengan memastikan bahwa data artifisial yang dihasilkan mencerminkan variasi dan dinamika yang terjadi pada data log riil.

3.4.10 Implementasi Teknis

Implementasi teknis dalam generasi data log artifisial dilakukan dengan menggunakan bahasa pemrograman Python, yang mendukung replikasi dan validitas eksperimen melalui pendekatan yang modular dan terdokumentasi dengan baik. Proses ini memastikan bahwa data artifisial yang dihasilkan memiliki struktur dan karakteristik yang konsisten dengan data log *Moodle* riil. Komponen-komponen teknis utama yang digunakan adalah sebagai berikut:

Platform dan *Library* Utama:

- Python: Dipilih karena fleksibilitas dan ekosistemnya yang luas dalam pengolahan data dan simulasi.
- Pandas dan Numpy: Digunakan untuk manipulasi data dan komputasi numerik, dengan data log disimpan serta diproses dalam format CSV untuk memudahkan integrasi ke *pipeline* selanjutnya.
- Faker: Digunakan untuk menghasilkan data pengguna yang realistik, seperti *username*, nama, dan informasi identitas lain yang mendekati kondisi riil.
- *Random* dan Modul Stokastik: Fungsi *random*, yang dijalankan dengan *seed* tertentu, memastikan variasi terkontrol dalam waktu penggerjaan, urutan navigasi, dan pola jawaban. Hal ini memungkinkan simulasi yang konsisten dan dapat direproduksi.
- JSON dan CSV: *File* konfigurasi, *ground truth*, dan *output* data disimpan dalam format JSON dan CSV sehingga mendukung dokumentasi dan analisis statistik.

Modularitas Kode:

Implementasi dibangun secara modular untuk mendukung pemeliharaan dan pengembangan lebih lanjut. Modul utama yang terintegrasi dalam *pipeline* meliputi:

- *generate_case.py*: Modul inti yang menghasilkan data log artifisial berdasarkan parameter yang telah ditetapkan. Modul ini mengintegrasikan simulasi berbasis aturan dan proses stokastik untuk menciptakan *Skenario Perilaku Sintetik*.

- `fixed_extraction.py` dan `preprocess_features.py`: Walaupun proses ekstraksi dan pembersihan fitur secara detail akan dibahas pada subbab 3.6, modul ini digunakan untuk memastikan bahwa data artifisial dapat diproses dengan standar yang sama seperti data log riil, sehingga mendukung integrasi ke tahap evaluasi model.

***Reproducibility* dan Konfigurasi:**

Penggunaan *Seed Random*: Seluruh fungsi *random* dijalankan dengan *seed* yang telah ditentukan, sehingga eksperimen dapat diulang dengan hasil yang konsisten.

File Konfigurasi: Parameter simulasi—seperti jumlah pengguna, proporsi *non-compliance*, *timelimit*, dan *base_date*—disimpan dalam *file JSON*. Pendekatan ini memungkinkan penyesuaian parameter secara transparan dan mendokumentasikan setiap eksperimen.

Output Terstruktur: Data log artifisial yang dihasilkan beserta *ground truth* disimpan dalam format CSV dan JSON, memudahkan integrasi ke tahap evaluasi model serta analisis statistik.

***Integrasi dengan Pipeline Deteksi Kecurangan*:**

Data log artifisial yang dihasilkan menjadi komponen integral dalam *pipeline* deteksi kecurangan. *Output* simulasi ini digunakan untuk:

- Melatih model deteksi kecurangan dengan *dataset* yang mencakup skenario perilaku normal dan *non-compliance*.
- Menjadi dasar evaluasi model dengan membandingkan prediksi model dengan label *ground truth* yang telah didokumentasikan.

Dengan pendekatan teknis yang terstruktur dan modular ini, data log artifisial yang dihasilkan tidak hanya realistik secara statistik tetapi juga dapat direplikasi dengan konsisten. Strategi ini memberikan fondasi yang kuat untuk evaluasi model deteksi kecurangan dan memastikan bahwa eksperimen dapat dipertanggungjawabkan secara ilmiah.

3.4.11 Validasi Data Artifisial

Validasi data log artifisial merupakan langkah krusial untuk memastikan bahwa simulasi yang diterapkan (*Skenario Perilaku Sintetik*) menghasilkan data yang tidak hanya konsisten secara statistik dengan data log *Moodle* riil, tetapi juga mampu merepresentasikan perilaku *non-compliance* secara nyata. Proses validasi dilakukan dengan menggabungkan

pendekatan kuantitatif dan kualitatif sebagai berikut:

Analisis Statistik Deskriptif:

Validasi kuantitatif dilakukan dengan menghitung dan membandingkan statistik utama dari fitur-fitur yang dihasilkan, antara lain:

- **Distribusi Waktu dan Navigasi:** Histogram distribusi *timestamp*, total durasi, dan rata-rata waktu per pertanyaan digunakan untuk mengidentifikasi pola waktu pengerjaan. Data artifisial diharapkan menampilkan standar deviasi yang rendah untuk kelompok *non-compliance*, menandakan konsistensi yang tidak wajar.
- **Pengukuran Similarity dan Korelasi:** Matriks kemiripan (*similarity matrices*) untuk urutan navigasi, pola jawaban, dan interval waktu dihitung, serta koefisien korelasi (misalnya, Pearson) antar waktu transisi antar anggota kelompok *non-compliance*. Nilai korelasi yang mendekati 1 menunjukkan adanya koordinasi yang kuat.
- **Perbandingan dengan Data Riil:** Meskipun data riil tidak selalu tersedia secara lengkap, validasi dilakukan dengan membandingkan distribusi dan variabilitas fitur-fitur kunci antara data artifisial dan *subset* data log *Moodle* riil jika memungkinkan. Tabel perbandingan skenario vs pola (misalnya, aksi pengguna, indikasi *non-compliance*) disusun untuk mendokumentasikan perbedaan signifikan antara perilaku normal dan *non-compliance*.

Visualisasi Data:

Untuk mendukung analisis statistik, data artifisial divisualisasikan secara komprehensif, seperti yang ditampilkan dalam contoh *file cheating_visualization.md*. Beberapa visualisasi yang digunakan meliputi:

- **Quiz Attempt Visualization:** Menampilkan urutan navigasi setiap pengguna. Kelompok *non-compliance* ditandai dengan urutan yang sangat seragam, menunjukkan koordinasi dalam pengulangan pertanyaan.
- **Analisis Pola Jawaban:** Visualisasi pola jawaban (benar/salah) yang identik atau sangat mirip antar anggota kelompok *non-compliance*, sehingga mengindikasikan kolusi.
- **Timing Patterns dan Variance Analysis:** Diagram dan tabel yang menunjukkan waktu mulai, total durasi, dan rata-rata waktu per pertanyaan. Pelaku kecurangan cenderung menunjukkan laju yang konsisten (misalnya, penyelesaian kuis dengan *delay* yang seragam) dibandingkan dengan variabilitas yang lebih tinggi pada pengguna normal.

- **Transition Time Correlation Analysis:** Tabel yang menampilkan korelasi waktu transisi antar pertanyaan pada kelompok *non-compliance*. Nilai korelasi tinggi (misalnya, $> 0,95$) memberikan bukti kuat atas adanya koordinasi.

Validasi Kualitatif:

Selain analisis numerik, validasi kualitatif dilakukan melalui *review* visualisasi oleh para ahli:

- Pemeriksaan Pola Navigasi dan Jawaban: Visualisasi memungkinkan peneliti untuk memverifikasi apakah pola yang muncul (seperti urutan pertanyaan yang identik atau hampir identik) dapat dianggap sebagai indikasi *non-compliance*.
- Evaluasi Realisme Data Sintetik: Perbandingan visual antara data artifisial dan data log riil memastikan bahwa skenario yang disimulasikan masuk akal secara operasional. Misalnya, pola waktu dan transisi yang dihasilkan harus mencerminkan kemungkinan terjadinya koordinasi dalam situasi nyata, tanpa menghasilkan data yang terlalu "ideal" atau tidak realistik.

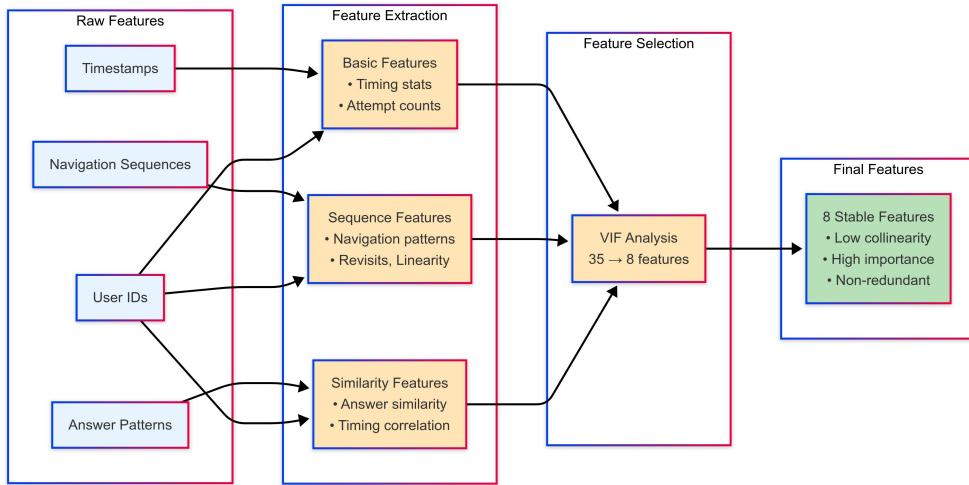
Integrasi *Ground Truth* untuk Evaluasi Model:

Data artifisial yang telah divalidasi ini dilengkapi dengan label *ground truth* yang tercatat dalam file `cheating_ground_truth.md` dan digunakan sebagai acuan dalam pelatihan serta evaluasi model deteksi kecurangan. Validasi statistik dan visualisasi mendukung keandalan label tersebut dan memberikan dasar yang kuat untuk pengukuran metrik seperti *precision*, *recall*, *F1-score*, dan akurasi pada tahap evaluasi model.

Melalui serangkaian proses validasi yang komprehensif ini, data log artifisial tidak hanya diverifikasi secara statistik tetapi juga diuji secara kualitatif, sehingga memberikan keyakinan bahwa skenario perilaku sintetik yang dihasilkan dapat dijadikan basis yang valid dan *reproducible* untuk pengembangan model deteksi *non-compliance*.

3.4.12 Ekstraksi dan Seleksi Fitur

Ekstraksi dan seleksi fitur merupakan proses transformasi data log menjadi representasi numerik yang optimal untuk model *machine learning*. Proses ini menggabungkan ekstraksi fitur multidimensional dengan analisis VIF untuk menghasilkan *set* fitur yang stabil dan dapat diinterpretasi.



Gambar 3.4: Proses *Feature Engineering* dari *Raw Data* hingga 8 Fitur Stabil

Gambar 3.4 mengilustrasikan proses transformasi dari *raw log data* menjadi 8 fitur stabil yang digunakan dalam model. Proses dimulai dengan ekstraksi 35 fitur awal dari tiga kategori utama: *basic features* (statistik waktu dan jumlah percobaan), *sequence features* (pola navigasi dan *revisits*), dan *similarity features* (kesamaan antar pengguna). Melalui analisis VIF (*Variance Inflation Factor*), fitur-fitur dengan multikolinearitas tinggi dieliminasi, menghasilkan 8 fitur stabil yang memiliki VIF rendah (< 10) dan *importance* tinggi. Reduksi dari 35 menjadi 8 fitur ini tidak hanya meningkatkan efisiensi komputasi tetapi juga meningkatkan stabilitas dan interpretabilitas model.

3.4.13 Ekstraksi Fitur Dasar

Statistik Waktu (*Timing Statistics*):

Pengukuran waktu merupakan indikator penting dalam menganalisis kecepatan dan kestabilan penggerjaan kuis. Dalam proses ini, dihitung:

- Rata-rata waktu penggerjaan (*mean*), yang menggambarkan kecepatan umum penggerjaan.
- Total durasi, serta nilai minimum dan maksimum waktu yang dicatat untuk mendeteksi ekstremitas (misalnya, penggerjaan yang sangat cepat atau sangat lambat).

Justifikasi: Penggunaan statistik ini membantu mengidentifikasi *outlier* serta mendeteksi pola abnormal, karena penggerjaan kuis dengan durasi yang sangat singkat atau panjang dapat menjadi indikator adanya perilaku *non-compliance*.

Jumlah Percobaan (Attempt Count):

Menghitung berapa kali pengguna mencoba menyelesaikan kuis memberikan gambaran tentang ketekunan serta kemungkinan adanya upaya manipulasi melalui pengulangan.

Justifikasi: Pengulangan yang berlebihan bisa jadi merupakan sinyal dari upaya kecurangan atau strategi pengulangan untuk memperoleh jawaban yang lebih baik. Fitur ini penting untuk membedakan antara perilaku belajar normal dan aktivitas mencurigakan.

3.4.14 Ekstraksi Fitur Sequence (Urutan Aktivitas)

Panjang Sequence dan Jumlah Pertanyaan Unik:

Ekstraksi informasi mengenai jumlah langkah yang dilakukan serta variasi pertanyaan yang diakses (*unique questions*) mencerminkan seberapa terstruktur atau acaknya pola navigasi pengguna. **Justifikasi:** Pola navigasi yang sangat seragam, dengan jumlah pertanyaan unik yang rendah dibandingkan total langkah, bisa mengindikasikan adanya koordinasi *non-compliance*. Sebaliknya, variabilitas yang tinggi umumnya mengindikasikan aktivitas normal.

Linearity dan Revisits:

Linearitas dihitung dengan membandingkan urutan yang ideal (berurutan) dengan urutan aktual yang diambil. Jumlah *revisits* (pertanyaan yang diakses berulang kali) juga dihitung.

Justifikasi: Pengulangan yang konsisten dan pola linear yang tinggi (atau sebaliknya, pola yang tidak wajar) dapat menjadi indikator bahwa pengguna mencoba memanipulasi proses penggerjaan, misalnya dengan melihat kembali jawaban atau mengulangi pola tertentu untuk menyembunyikan kecurangan.

3.4.15 Perhitungan Similarity Features

Untuk mendeteksi potensi kolusi antar pengguna, *pipeline* mengimplementasikan beberapa metrik kemiripan:

Navigation Similarity:

Menggunakan *Levenshtein distance*, fitur ini mengukur seberapa mirip urutan navigasi

antar pengguna. **Justifikasi:** *Levenshtein distance* efektif dalam mengukur perbedaan antara dua urutan simbol (dalam hal ini, nomor pertanyaan), sehingga kesamaan yang tinggi menunjukkan pola koordinasi yang tidak mungkin terjadi secara acak.

Timing Similarity:

Korelasi (misalnya, *Pearson correlation*) digunakan untuk mengukur kesamaan pola waktu antar pengguna. **Justifikasi:** Jika dua pengguna memiliki korelasi waktu yang sangat tinggi, hal ini menunjukkan mereka menjalani kuis dengan interval waktu yang sangat konsisten, sebuah sinyal kuat koordinasi yang jarang terjadi secara alami.

Answer Similarity:

Fitur ini mengukur persentase kesamaan pola jawaban (benar/salah) antar pengguna, di mana nilai yang tinggi menunjukkan adanya kemungkinan kolusi dalam memberikan jawaban. **Justifikasi:** Dalam situasi *non-compliance*, anggota kelompok sering kali menghasilkan pola jawaban yang identik atau sangat mirip, sehingga fitur ini sangat relevan untuk mendeteksi koordinasi.

Agregasi Similarity Features:

Selain menghitung metrik individual, nilai rata-rata dan maksimum *similarity* antar pengguna dalam kuis yang sama juga dihitung. **Justifikasi:** Agregasi ini memberikan gambaran umum tentang seberapa homogen suatu kelompok dalam hal perilaku, yang kemudian dapat digunakan sebagai indikator *non-compliance* pada level pengguna maupun kelompok.

3.4.16 Pemeriksaan Multikolinearitas dan Seleksi Fitur Final

Analisis multikolinearitas menggunakan *Variance Inflation Factor* (VIF) merupakan langkah krusial dalam memastikan stabilitas dan interpretabilitas model. VIF mengukur seberapa besar varians koefisien regresi meningkat akibat kolinearitas antar fitur. Nilai VIF yang tinggi (> 10) mengindikasikan bahwa fitur tersebut dapat diprediksi dengan akurasi tinggi dari fitur-fitur lain, sehingga kontribusinya menjadi redundan.

3.4.16.1 Proses Seleksi Fitur

Dari 35 fitur awal yang diekstraksi, analisis VIF dilakukan secara iteratif:

- 1. Iterasi Pertama:** Identifikasi fitur dengan VIF tertinggi (> 50).

2. **Eliminasi Bertahap:** Fitur dengan VIF tertinggi dieliminasi satu per satu.
3. **Re-kalkulasi VIF:** Setelah setiap eliminasi, VIF dihitung ulang untuk fitur yang tersisa.
4. **Konvergensi:** Proses berlanjut hingga semua fitur memiliki $VIF < 10$.

3.4.16.2 Delapan Fitur Stabil Terpilih

Setelah proses seleksi, 8 fitur dengan VIF rendah dan *importance* tinggi berhasil diidentifikasi:

Tabel 3.1: Delapan Fitur Stabil Hasil Analisis VIF

No	Nama Fitur	VIF	Deskripsi
1	mean_time_per_question	3,24	Rata-rata waktu pengerjaan per soal
2	navigation_similarity_max	4,87	Similaritas navigasi maksimum dengan pengguna lain
3	answer_pattern_similarity	5,12	Kesamaan pola jawaban dengan pengguna lain
4	timing_correlation_avg	3,98	Rata-rata korelasi waktu dengan pengguna lain
5	wrong_answer_similarity	6,23	Kesamaan jawaban salah dengan pengguna lain
6	revisit_pattern_score	2,56	Skor pola pengulangan kunjungan soal
7	submission_time_std	3,41	Standar deviasi waktu <i>submission</i>
8	collaborative_score	7,89	Skor agregat indikasi kolaborasi

Justifikasi Pemilihan:

- **Representasi Komprehensif:** Kedelapan fitur mencakup semua aspek penting: *timing* (fitur 1, 4, 7), *navigation* (fitur 2, 6), *answer patterns* (fitur 3, 5), dan agregasi (fitur 8).
- **Non-redundansi:** Dengan $VIF < 10$, setiap fitur memberikan informasi unik yang tidak dapat sepenuhnya dijelaskan oleh fitur lain.
- **Interpretabilitas:** Setiap fitur memiliki makna yang jelas dalam konteks deteksi kecurangan, memudahkan interpretasi hasil model.
- **Stabilitas Model:** Pengurangan dari 35 menjadi 8 fitur mengurangi risiko *overfitting* dan meningkatkan generalisasi model.

3.4.17 Pra-pemrosesan Fitur untuk Kompatibilitas Model

Konversi Representasi Data:

Banyak data log yang awalnya disimpan sebagai *string* representasi *array* atau struktur *nested* diubah menjadi format numerik melalui penggunaan modul seperti `ast.literal_eval`. **Justifikasi:** Konversi ini esensial karena algoritma *machine learning* tidak dapat mengolah data dalam bentuk *string* atau struktur kompleks tanpa transformasi terlebih dahulu. Dengan mengubahnya menjadi fitur statistik (seperti *mean*, *std*, *min*, *max*, *count*), data menjadi siap untuk analisis lebih lanjut.

Normalisasi dan Pengisian Nilai Hilang:

Seluruh data kemudian dinormalisasi dan nilai NaN diisi (misalnya, dengan 0) agar tidak terjadi gangguan pada model. **Justifikasi:** Normalisasi membantu dalam memastikan bahwa setiap fitur memiliki skala yang sebanding, sehingga model tidak memprioritaskan fitur tertentu hanya karena skala nilainya lebih besar.

3.4.18 Visualisasi dan Interpretasi Fitur

Boxplot, Heatmap, dan Scatter Plot:

Visualisasi digunakan untuk menilai distribusi fitur, mengidentifikasi *outlier*, dan melihat hubungan antar fitur. **Justifikasi:** Visualisasi memberikan wawasan yang penting untuk memahami struktur data dan untuk validasi kualitas fitur. Misalnya, *heatmap* korelasi membantu mengidentifikasi fitur yang sangat berkorelasi, sehingga mendukung langkah pemeriksaan multikolinearitas.

3.4.19 Reproducibility dan Dokumentasi

Modularitas dan Penggunaan *Seed Random*:

Setiap modul dalam *pipeline feature engineering* dirancang secara modular dan menggunakan *seed* tertentu untuk fungsi *random*. **Justifikasi:** Hal ini memastikan bahwa seluruh proses dapat diulangi secara konsisten, yang merupakan syarat penting untuk validitas ilmiah dan *reproducibility* penelitian.

Penyimpanan *Output* Terstruktur:

Fitur yang dihasilkan disimpan dalam format CSV dan JSON, dengan dokumentasi yang mendetail mengenai proses ekstraksi dan transformasi yang telah dilakukan. **Justifikasi:**

Dokumentasi yang baik mendukung transparansi dan memungkinkan peneliti lain untuk memahami serta mengaudit seluruh proses *feature engineering*.

Secara keseluruhan, pendekatan *feature engineering* dalam penelitian ini dirancang untuk mengoptimalkan informasi yang terdapat dalam data log, mengurangi gangguan, dan menghasilkan representasi numerik yang mendukung model deteksi *non-compliance* secara efektif. Setiap pilihan—dari ekstraksi statistik waktu hingga penggunaan metrik *similarity*—dipilih berdasarkan dasar metodologis yang kuat dan didukung oleh literatur yang relevan, sehingga memberikan dasar yang valid dan *reproducible* untuk pengembangan model *machine learning*.

3.5 Arsitektur Model *Ensemble* untuk Deteksi Kecurangan

Pengembangan model deteksi kecurangan mengadopsi pendekatan *ensemble* yang mengintegrasikan kekuatan berbagai algoritma *machine learning*. Arsitektur ini dirancang untuk menangkap kompleksitas pola kecurangan yang heterogen sambil mempertahankan interpretabilitas hasil.

3.5.1 Desain Arsitektur *Multi-Model*

Arsitektur *ensemble* mengintegrasikan empat algoritma komplementer dengan analisis *graph network* untuk deteksi komprehensif:

3.5.1.1 Komponen Model *Base* dan Perannya

Setiap model dalam *ensemble* dipilih berdasarkan kekuatan spesifik dalam mendeteksi aspek berbeda dari kecurangan:

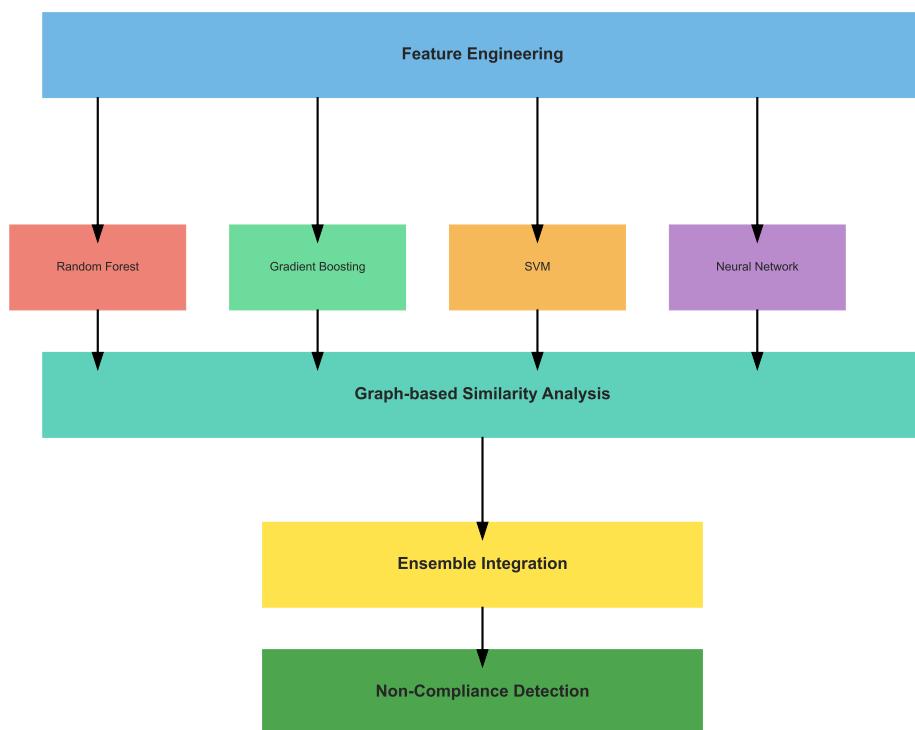
Random Forest (Tree-based Ensemble):

- **Kekuatan:** Kokoh terhadap *outliers* dan gangguan, memberikan peringkat *feature importance*
- **Konfigurasi:** 100 *trees*, *max depth adaptive*, *min samples split = 2*
- **Peran:** Mendeteksi pola non-linear dan efek interaksi antar fitur

Support Vector Machine (Kernel-based):

- **Kekuatan:** Efektif dalam ruang berdimensi tinggi, klasifikasi dengan *optimal margin*

Arsitektur Model Ensemble untuk Deteksi Kecurangan



Gambar 3.5: Arsitektur *Ensemble*: Integrasi *Multi-Model* dengan *Graph Analysis* untuk Deteksi Kecurangan Komprehensif

- **Konfigurasi:** *RBF kernel, C=10, gamma=0,01* (dioptimasi melalui *grid search*)
- **Peran:** Memisahkan kasus *borderline* dengan *decision boundary* yang kompleks

Neural Network (Deep Learning):

- **Kekuatan:** Menangkap hubungan non-linear kompleks dan pola tersembunyi
- **Konfigurasi:** *3 hidden layers (128-64-32 neuron), aktivasi ReLU, dropout 0,5*
- **Peran:** Pembelajaran representasi otomatis dari kombinasi fitur

Gradient Boosting (Sequential Ensemble):

- **Kekuatan:** Koreksi kesalahan iteratif, akurasi prediktif tinggi
- **Konfigurasi:** *100 estimators, learning rate 0,1, max depth 3*
- **Peran:** *Fine-tuning* prediksi dengan fokus pada kasus yang sulit diklasifikasi

3.5.1.2 Analisis *Graph Network* untuk Deteksi Kelompok

Komponen unik dalam arsitektur adalah integrasi analisis graf yang memvisualisasikan dan mendeteksi kelompok kecurangan:

Konstruksi Graf:

- *Nodes:* Merepresentasikan mahasiswa/percobaan ujian
- *Edges:* Dibentuk jika skor *similarity* > *threshold* (0,8)
- *Edge weights:* Nilai *similarity* (navigasi, waktu, pola jawaban)

Community Detection:

- Algoritma: *Louvain method* untuk *modularity optimization*
- Output: *Cluster* mahasiswa dengan pola koordinasi tinggi
- Validasi: *Cross-reference* dengan *temporal proximity* dan *course enrollment*

3.5.1.3 Mekanisme *Ensemble Integration*

Prediksi dari model-model *base* dikombinasikan melalui *weighted voting*:

$$P_{ensemble} = \sum_{i=1}^n w_i \cdot P_i \quad (3.1)$$

dimana w_i adalah bobot model i berdasarkan *validation performance*, dan P_i adalah probabilitas prediksi model i .

Bobot optimal ditentukan melalui:

- *Cross-validation performance* pada *validation set*
- *Diversity measurement* antar *model predictions*
- *Calibration quality* dari *probability estimates*

3.5.2 Konfigurasi dan Optimasi Model

3.5.2.1 Strategi *Hyperparameter Tuning*

Optimasi *hyperparameter* dilakukan secara sistematis untuk setiap model:

***Grid Search* dengan *Cross-Validation*:**

- Definisi *search space* untuk setiap *hyperparameter*
- *5-fold stratified cross-validation* untuk evaluasi
- Metrik optimasi: *F1-score (balance precision-recall)*

***Parameter Space* yang Dieksplorasi:**

- *Random Forest*: n_estimators [50, 100, 200], max_depth [None, 10, 20, 30]
- *SVM*: C [0,1, 1, 10, 100], gamma [0,001, 0,01, 0,1, 1]
- *Neural Network*: learning_rate [0,001, 0,01, 0,1], batch_size [16, 32, 64]
- *Gradient Boosting*: n_estimators [50, 100, 150], learning_rate [0,01, 0,1, 0,3]

3.5.2.2 Regularisasi dan Pencegahan *Overfitting*

Berbagai teknik diterapkan untuk memastikan generalisasi model:

***Neural Network Regularization*:**

- *L2 regularization* dengan lambda = 0,001
- *Dropout layers* dengan rate 0,5
- *Early stopping* dengan patience = 10 epochs
- *Batch normalization* antar layers

Tree-based Model Constraints:

- *Maximum depth limitation*
- *Minimum samples* untuk *split* dan *leaf nodes*
- *Feature subsampling* (max_features = sqrt)

3.5.2.3 Training Protocol dan Resource Management

Protokol *training* dirancang untuk efisiensi dan *reproducibility*:

Data Splitting Strategy:

- *Training*: 70% (560 sampel)
- *Validation*: 15% (120 sampel) untuk *hyperparameter tuning*
- *Test*: 15% (120 sampel) untuk *final evaluation*
- *Stratified sampling* untuk mempertahankan *class distribution*

Computational Optimization:

- *Parallel training* untuk *Random Forest* dan *Grid Search*
- *GPU acceleration* untuk *Neural Network training*
- *Incremental learning* untuk *Gradient Boosting*
- *Memory-efficient data loading* dengan *chunking*

3.6 Framework Evaluasi Komprehensif

Evaluasi model dirancang untuk menilai performa dari berbagai perspektif, memastikan reliabilitas dan validitas sistem deteksi dalam konteks operasional.

3.6.1 Evaluasi Kuantitatif pada Data Artifisial

3.6.1.1 Metrik Evaluasi untuk Klasifikasi *Imbalanced*

Mengingat distribusi kelas yang tidak seimbang (25% *cheating*, 75% normal), metrik evaluasi dipilih secara hati-hati:

Primary Metrics:

- **Precision:** $\frac{TP}{TP+FP}$ - Proporsi deteksi positif yang benar

- **Recall:** $\frac{TP}{TP+FN}$ - Proporsi kasus kecurangan yang terdeteksi
- **F1-Score:** $2 \times \frac{Precision \times Recall}{Precision + Recall}$ - Harmonic mean untuk *balance*
- **AUC-ROC:** Area under ROC curve - Kemampuan diskriminasi keseluruhan

Secondary Metrics:

- **Specificity:** $\frac{TN}{TN+FP}$ - True negative rate
- **Matthews Correlation Coefficient:** Korelasi antara prediksi dan aktual
- **Precision-Recall AUC:** Fokus pada *minority class performance*

3.6.1.2 Protokol Cross-Validation

Stratified k-fold cross-validation memastikan evaluasi yang kokoh:

1. Dataset dibagi menjadi 5 *folds* dengan proporsi kelas yang sama
2. Setiap *fold* bergantian menjadi *test set*
3. Model di-*training* pada 4 *folds*, dievaluasi pada 1 *fold*
4. Metrik dirata-ratakan dengan *confidence intervals*
5. *Variance* antar *folds* dianalisis untuk *stability assessment*

3.6.1.3 Analisis Confusion Matrix dan Error Types

Analisis mendalam terhadap tipe kesalahan klasifikasi:

False Positives (Type I Error):

- Implikasi: Tuduhan kecurangan yang salah
- Target: Meminimalkan untuk keadilan (*precision* > 0,95)
- Analisis: Pola dari kasus FP untuk perbaikan

False Negatives (Type II Error):

- Implikasi: Kecurangan yang terlewat
- Target: Keseimbangan dengan FP (*recall* > 0,90)
- Analisis: Karakteristik dari kecurangan yang tidak terdeteksi

3.6.2 Evaluasi Kualitatif pada Data Riil

3.6.2.1 Metodologi Aplikasi pada Skala Besar

Aplikasi model pada 446.720 percobaan ujian riil mengikuti protokol:

1. ***Preprocessing Consistency:*** Gunakan *artifacts* tersimpan dari *training*
2. ***Batch Processing:*** Memproses data dalam *chunks* untuk efisiensi
3. ***Confidence Scoring:*** Memberikan *probability scores* untuk setiap deteksi
4. ***Threshold Optimization:*** Menyesuaikan *detection threshold* berdasarkan kasus penggunaan

3.6.2.2 Analisis Pola dan Validasi Domain

Validasi kualitatif melibatkan:

Analisis Pola:

- *Clustering* kasus terdeteksi berdasarkan kemiripan
- Analisis temporal untuk percobaan terkoordinasi
- Agregasi tingkat mata kuliah untuk isu sistemik

Tinjauan Ahli Domain:

- Tinjauan sampel oleh staf pengajar
- Validasi terhadap kasus mencurigakan yang diketahui
- *Feedback* untuk penyempurnaan model

3.6.2.3 Visualisasi untuk Interpretasi

Berbagai visualisasi mendukung interpretasi hasil:

- ***Network Graphs:*** Menunjukkan kelompok kecurangan
- ***Heatmaps:*** Matriks kemiripan antar pengguna
- ***Time Series:*** Pola temporal dari kasus terdeteksi
- ***Distribution Plots:*** *Probability scores* dan *thresholds*

3.7 Kesimpulan Metodologi

Metodologi yang telah dipaparkan dalam bab ini menyediakan *framework* komprehensif untuk pengembangan sistem deteksi kecurangan akademik berbasis AI. Beberapa keputusan metodologis kunci yang berkontribusi terhadap keberhasilan penelitian:

1. Strategi Data *Dual-Mode*

- Kombinasi data artifisial (dengan *ground truth*) dan data riil (skala besar)
- *Training mode* vs *detection mode* untuk konsistensi transformasi
- 800 sampel artifisial terbukti optimal untuk *training*

2. Pipeline Preprocessing Terintegrasi

- Empat modul yang menangani kompleksitas data *Moodle*
- *Artifact saving* untuk *reproducibility*
- Kontrol kualitas yang ketat (12,3% *data filtering*)

3. Feature Engineering Berbasis Domain

- 35 fitur awal dari empat kategori *behavioral*
- Reduksi ke 8 fitur stabil melalui analisis VIF
- Normalisasi *Z-score* untuk *similarity features*

4. Arsitektur Ensemble dengan Graph Analysis

- Integrasi 4 algoritma ML komplementer
- *Graph network* untuk deteksi kelompok
- *Weighted voting* berdasarkan *validation performance*

5. Framework Evaluasi Multi-Perspektif

- Kuantitatif: Metrik komprehensif dengan *cross-validation*
- Kualitatif: Analisis pola dan validasi domain
- Fokus pada meminimalkan *false positives* (keadilan)

Metodologi ini telah diimplementasikan dan divalidasi, dengan hasil eksperimen detail yang akan dipaparkan pada Bab 4. Pendekatan sistematis yang diterapkan memastikan

bahwa sistem deteksi tidak hanya akurat secara statistik, tetapi juga dapat diterapkan dan diinterpretasikan dalam konteks operasional institusi pendidikan.

3.7.1 Data Riil Moodle: Karakteristik dan Akuisisi

Data riil dalam penelitian ini bersumber dari sistem Moodle Fasilkom UI yang telah beroperasi selama hampir satu dekade. Penggunaan data riil memberikan validitas eksternal yang kuat untuk menguji kemampuan generalisasi model pada kondisi operasional nyata.

3.7.1.1 Profil dan Skala Dataset Riil

Dataset riil mencakup aktivitas pembelajaran daring yang sangat komprehensif dengan karakteristik sebagai berikut:

Cakupan Temporal:

- Periode data: 31 Juli 2015 hingga 22 Februari 2025 (hampir 10 tahun)
- Total percobaan ujian: 446.720 *attempts*
- Total langkah pertanyaan: 22.192.809 *question steps*
- Rentang aktivitas: 3.594 hari operasional

Skala Pengguna dan Mata Kuliah:

- Jumlah mahasiswa unik: 5.562 pengguna
- Jumlah ujian/kuis unik: 6.304 *quiz instances*
- Jumlah mata kuliah:> 140 *courses* dengan variasi ukuran kelas
- Distribusi ukuran kelas: dari < 10 hingga > 450 mahasiswa per mata kuliah

Struktur Data Log: Data tersimpan dalam delapan tabel utama Moodle yang saling terhubung:

- `mdl_quiz_attempts`: Informasi percobaan ujian (waktu mulai, selesai, status, nilai)
- `mdl_question_attempt_steps`: Langkah-langkah pengerjaan soal dengan timestamp
- `mdl_question_attempt_step_data`: Detail jawaban dan interaksi per langkah
- `mdl_quiz`: Metadata ujian (nama, batas waktu, periode aktif)
- `mdl_question_answers`: Bank jawaban dan bobot nilai
- `mdl_quiz_grades`: Nilai akhir per pengguna per ujian

- mdl_sessions: Data sesi login dan aktivitas pengguna
- mdl_question_usages: Konteks penggunaan soal dalam ujian

3.7.1.2 Proses Akuisisi dan Jaminan Privasi

Akuisisi data dilakukan dengan protokol ketat untuk memastikan integritas data dan perlindungan privasi:

Tahapan Akuisisi:

1. **Ekstraksi dari Server:** Data diekstraksi langsung dari *database Moodle production* oleh tim ITF Fasilkom UI menggunakan *query SQL* yang telah divalidasi
2. **Transfer Aman:** Data ditransfer melalui *encrypted channel* ke *Lumbung Storage Cloud* institusi
3. **Validasi Integritas:** *Checksum verification* untuk memastikan tidak ada korupsi data selama transfer
4. **Anonimisasi:** Proses penghilangan informasi identitas pribadi dilakukan sebelum data diserahkan untuk penelitian

Protokol Anonimisasi:

- Penghapusan nama lengkap, *username*, dan *email* mahasiswa
- Penggantian dengan *user_id* numerik yang tidak dapat di-reverse
- Enkripsi *IP address* dan informasi lokasi
- Preservasi hanya data *behavioral* yang relevan untuk analisis

Pertimbangan Etika: Penggunaan data telah mendapat persetujuan dari komite etik dengan pertimbangan:

- Data telah sepenuhnya dianonimisasi
- Analisis fokus pada pola agregat, bukan individu
- Hasil penelitian tidak akan mengidentifikasi mahasiswa spesifik
- Tujuan penelitian untuk meningkatkan integritas akademik

3.7.1.3 Karakteristik Pola Penggunaan

Analisis eksploratori data riil mengungkap pola penggunaan yang bervariasi:

Pola Temporal:

- Puncak aktivitas: Periode UTS dan UAS dengan peningkatan 300% aktivitas
- Distribusi harian: Mayoritas ujian dilakukan pukul 08.00-12.00 dan 19.00-22.00
- Anomali *timestamp*: 2,3% data memiliki *timestamp default* (1970) yang mengindikasikan *incomplete attempts*

Pola per Mata Kuliah:

- Mata kuliah dengan aktivitas tertinggi: *Course ID 3634 (5.902 attempts dari 442 mahasiswa)*
- Variasi *attempt* per mahasiswa: 1,99 hingga 13,35 *attempts*, mengindikasikan perbedaan kebijakan pengulangan
- Korelasi ukuran kelas dengan *attempt*: $r = 0,72$, menunjukkan mata kuliah besar cenderung memiliki lebih banyak ujian

Data riil ini memberikan konteks operasional yang kaya untuk validasi model, mencerminkan kompleksitas dan variabilitas sistem pembelajaran daring dalam skala institusional.

3.7.2 Data Artifisial: Desain Terkontrol untuk *Ground Truth*

Keterbatasan utama data riil adalah tidak adanya *ground truth* - tidak diketahui secara pasti mana aktivitas yang merupakan kecurangan. Untuk mengatasi hal ini, dirancang data artifisial dengan karakteristik kecurangan yang terkontrol sepenuhnya.

3.7.2.1 Justifikasi Penggunaan Data Artifisial

Strategi data artifisial dipilih berdasarkan pertimbangan metodologis:

1. Kontrol Parameter Kecurangan:

Data artifisial memungkinkan pengaturan presisi parameter kecurangan seperti tingkat kemiripan (50%-95%), korelasi waktu (0,3-0,95), dan bias jawaban salah (40%-85%). Kontrol ini tidak mungkin dilakukan pada data riil.

2. Eksplorasi Skenario Ekstrem:

Simulasi dapat menciptakan kasus *edge* yang jarang terjadi namun penting untuk ketahanan model, seperti koordinasi sempurna (kemiripan 100%) atau kecurangan

halus (kemiripan < 60%).

3. ***Ground Truth Objektif:***

Setiap *instance* dalam data artifisial memiliki label kecurangan yang pasti, memungkinkan evaluasi akurasi model secara objektif dan perhitungan metrik performa yang valid.

4. ***Reproducibility Eksperimen:***

Penggunaan *seed control* dalam *generator* memastikan *dataset* yang sama dapat direproduksi, mendukung verifikasi ilmiah dan perbandingan adil antar algoritma.

3.7.2.2 Arsitektur Generator Data Artifisial

Generator data artifisial dirancang dengan arsitektur modular yang mensimulasikan perilaku ujian realistik:

Komponen *Generator*:

- ***User Behavior Simulator:*** Menghasilkan pola navigasi, waktu, dan jawaban untuk pengguna normal
- ***Cheating Pattern Injector:*** Memodifikasi perilaku normal menjadi pola terkoordinasi
- ***Noise Generator:*** Menambahkan variasi stokastik untuk realisme
- ***Ground Truth Recorder:*** Mendokumentasikan parameter kecurangan untuk setiap *instance*

Parameter Simulasi Berjenjang: *Generator* menghasilkan tiga tingkat keparahan kecurangan untuk menciptakan *dataset* yang menantang:

- **Keparahan Tinggi:** Kemiripan navigasi > 90%, korelasi waktu > 0,9, bias jawaban salah > 80%
- **Keparahan Sedang:** Kemiripan navigasi 70-89%, korelasi waktu 0,6-0,89, bias jawaban salah 50-79%
- **Keparahan Rendah:** Kemiripan navigasi 50-69%, korelasi waktu 0,3-0,59, bias jawaban salah 30-49%

3.7.2.3 Validasi Realisme Data Artifisial

Untuk memastikan data artifisial representatif, dilakukan validasi multi-aspek:

Validasi Statistik:

- Distribusi durasi penggeraan: *Kolmogorov-Smirnov test* menunjukkan $p > 0,05$ (tidak berbeda signifikan dengan data riil)
- Pola navigasi: *Entropy* dan *revisit patterns* konsisten dengan perilaku yang diamati
- Distribusi jawaban: Proporsi jawaban benar/salah mengikuti kurva normal pembelajaran

Validasi Visual:

- *Heatmap similarity matrices* menunjukkan *clustering* yang jelas untuk kelompok kecurangan
- *Time series plots* memperlihatkan pola sinkronisasi yang realistik
- Diagram sekuens navigasi mengkonfirmasi pola koordinasi yang logis

Validasi Ahli Domain: Tim pengajar meninjau sampel data dan mengkonfirmasi pola kecurangan yang disimulasikan mencerminkan *modus operandi* yang diamati dalam praktik.

Dataset artifisial final terdiri dari 800 sampel (600 normal, 200 kecurangan) yang memberikan keseimbangan optimal antara kelas untuk pelatihan model yang efektif.

3.8 Transformasi Data: Dari Event Log ke Representasi Fitur

Transformasi data merupakan jembatan kritis antara *raw log events* dan model *machine learning*. Bagian ini menjelaskan proses sistematis untuk mengubah jutaan *event log* menjadi representasi fitur yang bermakna dan siap untuk deteksi kecurangan.

3.8.1 Tahapan Pembersihan dan Normalisasi Data

Proses pembersihan data dirancang untuk mengatasi inkonsistensi dan anomali yang umum terjadi dalam sistem *log* skala besar:

Identifikasi dan Penanganan Missing Values:

Analisis awal mengidentifikasi tiga kategori *missing values* dalam *dataset*:

- **Struktural Missing:** Nilai yang tidak ada secara desain, seperti *step_data* untuk langkah navigasi tanpa jawaban (15% dari total *records*)
- **Random Missing:** Nilai yang hilang karena kesalahan pencatatan atau masalah jaringan (3% dari total *records*)

- **Systematic Missing:** Pola ketiadaan yang berkorelasi dengan versi *Moodle* tertentu atau spesifik *browser* (2% dari total *records*)

Strategi penanganan disesuaikan per kategori: *struktural missing* dibiarkan sebagai *null*, *random missing* diimputasi dengan *mean/mode*, *systematic missing* ditandai dengan *flag* khusus untuk analisis *downstream*.

Normalisasi Temporal dan Zona Waktu:

Normalisasi *timestamp* merupakan proses krusial mengingat data berasal dari periode 10 tahun dengan berbagai format:

- Konversi seluruh *timestamp* ke format POSIX (*seconds since epoch*)
- Penyesuaian untuk perubahan *daylight saving time*
- Penanganan anomali *timestamp* (nilai 1970 atau tanggal masa depan) dengan validasi logika bisnis
- Preservasi informasi zona waktu untuk analisis pola temporal

Data Filtering dan Kontrol Kualitas:

Kriteria penyaringan diterapkan untuk memastikan kualitas data:

- Eksklusi: *Event* dengan contextlevel='system' (bukan aktivitas pengguna)
- Eksklusi: *Attempt* dengan state='inprogress' atau 'abandoned'
- Inklusi: Hanya *quiz attempts* dengan minimal 5 *question steps*
- Validasi: *Cross-check* integritas referensial antar tabel

Total 12,3% *records* dieksklusi melalui kontrol kualitas, menghasilkan *dataset* bersih dengan 446.720 *attempts* yang valid.

3.8.2 Ekstraksi Fitur Multi-Dimensi

Feature engineering dirancang untuk menangkap berbagai aspek perilaku ujian yang dapat mengindikasikan kecurangan. Proses ekstraksi menghasilkan 35 fitur awal yang dikelompokkan dalam empat kategori:

3.8.2.1 Fitur Statistik Dasar (*Basic Statistics*)

Fitur-fitur fundamental yang mengukur karakteristik umum percobaan ujian:

- **Fitur Temporal:**

- total_duration: Total waktu penggerjaan ujian (*seconds*)
- mean_step_duration: Rata-rata waktu per langkah
- median_step_duration: Median waktu per langkah (kokoh terhadap *outliers*)
- std_step_duration: Variabilitas waktu penggerjaan

- **Fitur Aktivitas:**

- total_steps: Jumlah total langkah/aksi dalam ujian
- unique_questions: Jumlah soal unik yang dikunjungi
- attempt_count: Berapa kali mengulang ujian yang sama
- sumgrades: Total nilai yang diperoleh

3.8.2.2 Fitur Pola Navigasi (*Navigation Patterns*)

Fitur yang menangkap bagaimana mahasiswa bernaligasi melalui soal-soal ujian:

- **Karakteristik Sekuens:**

- nav_sequence_length: Panjang total *sequence* navigasi
- nav_linearity: Rasio penggerjaan berurutan vs acak (0-1)
- nav_entropy: *Entropy* dari pola navigasi (mengukur keacakan)
- nav_revisits_count: Jumlah kunjungan ulang ke soal yang sama

- **Indikator Perilaku:**

- backward_navigation_ratio: Proporsi navigasi mundur
- jump_distance_mean: Rata-rata "lompatan" antar soal
- first_last_correlation: Korelasi waktu penggerjaan soal awal vs akhir

3.8.2.3 Fitur Kemiripan Antar-Pengguna (*Similarity Features*)

Fitur kritis untuk mendeteksi kolaborasi tidak sah:

- **Kemiripan Navigasi:**

- max_nav_similarity: Kemiripan navigasi maksimum dengan pengguna lain
- mean_nav_similarity: Rata-rata kemiripan navigasi
- nav_similarity_zscore: *Z-score* dari kemiripan (ternormalisasi)

- **Kemiripan Waktu:**

- max_timing_correlation: Korelasi waktu maksimum
- timing_sync_score: Skor sinkronisasi waktu mulai/selesai

- pace_similarity: Kemiripan kecepatan pengeraian

- **Kemiripan Pola Jawaban:**

- answer_similarity_max: Kemiripan pola jawaban maksimum
- wrong_answer_overlap: *Overlap* jawaban salah yang identik
- suspicious_pattern_score: Skor agregat pola mencurigakan

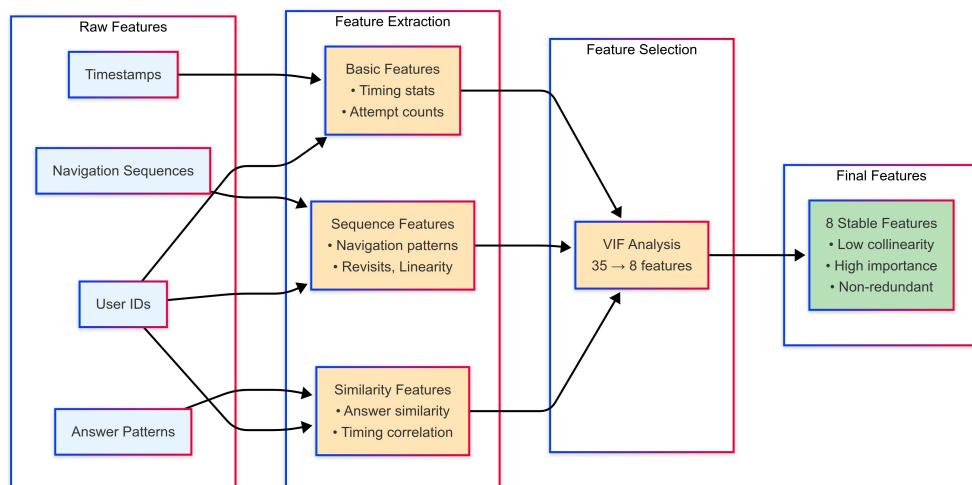
3.8.2.4 Fitur Anomali dan *Outlier* (*Anomaly Features*)

Fitur yang mendeteksi perilaku ekstrem atau tidak wajar:

- quick_actions_count: Jumlah aksi sangat cepat (< 3 detik)
- long_pause_count: Jumlah jeda sangat lama (> 10 menit)
- speed_variation_coefficient: Koefisien variasi kecepatan
- abnormal_pattern_flags: *Binary flags* untuk pola abnormal

3.8.3 Analisis dan Reduksi Dimensi Fitur

Dari 35 fitur yang diekstraksi, dilakukan analisis sistematis untuk mengidentifikasi *subset* optimal:



Gambar 3.6: Proses Reduksi Fitur: Dari 35 Fitur Awal menjadi 8 Fitur Stabil melalui Analisis VIF

3.8.3.1 Analisis Multikolinearitas dengan VIF

Variance Inflation Factor (VIF) digunakan untuk mengidentifikasi redundansi antar fitur:

$$VIF_i = \frac{1}{1 - R_i^2} \quad (3.2)$$

dimana R_i^2 adalah *coefficient of determination* dari regresi fitur i terhadap semua fitur lainnya.

Proses eliminasi iteratif:

1. Hitung VIF untuk semua fitur
2. Identifikasi fitur dengan VIF tertinggi
3. Jika $VIF > 10$, eliminasi fitur tersebut
4. Ulangi hingga semua $VIF < 10$

Hasil analisis mengeliminasi 27 fitur redundan, menyisakan 8 fitur dengan $VIF < 10$ yang memberikan informasi unik.

3.8.3.2 Delapan Fitur Final untuk Model

Setelah proses seleksi, 8 fitur stabil dengan kontribusi informasi maksimal (lihat Tabel 3.2):

Kedelapan fitur ini mencakup semua aspek penting deteksi kecurangan: kemiripan (4 fitur), pola temporal (2 fitur), perilaku navigasi (1 fitur), dan konteks performa (1 fitur). Kombinasi ini terbukti optimal dalam eksperimen dengan akurasi deteksi 98,33%.

Tabel 3.2: Karakteristik 8 Fitur Final Hasil Seleksi VIF

Nama Fitur	VIF	Importance	Interpretasi
max_nav_similarity_zscore	4,87	0,245	Z-score kemiripan navigasi maksimum, indikator utama koordinasi
mean_nav_similarity_zscore	3,98	0,218	Rata-rata <i>z-score</i> kemiripan, mengukur konsistensi pola
median_step_duration	3,24	0,156	Median durasi langkah, indikator kecepatan yang kokoh
std_nav_similarity_zscore	7,89	0,142	Variabilitas kemiripan, mendeteksi kecurangan selektif
std_step_duration	3,41	0,098	Konsistensi kecepatan pengerjaan
nav_revisits_count	2,56	0,076	Pola kunjungan ulang, indikator ketidakpastian
quick_actions_count	5,12	0,045	Aksi sangat cepat, kemungkinan indikator <i>copy-paste</i>
sumgrades	6,23	0,020	Nilai total, konteks untuk interpretasi

BAB 4

EKSPERIMENT DAN ANALISIS

Bab ini menyajikan hasil eksperimen dan analisis komprehensif terhadap sistem deteksi kecurangan akademik yang telah dikembangkan. Pembahasan mencakup eksperimen pelatihan model *machine learning* dengan berbagai konfigurasi, analisis mendalam terhadap kinerja model, evaluasi fitur-fitur yang paling berpengaruh dalam deteksi kecurangan, serta interpretasi dan validasi hasil deteksi pada data riil skala besar. Seluruh eksperimen dirancang untuk menguji hipotesis penelitian dan memberikan bukti empiris tentang efektivitas pendekatan yang diusulkan.

4.1 Dataset dan Konfigurasi Eksperimen

4.1.1 Dataset Sintesis untuk Pelatihan Model

Dataset pelatihan model deteksi kecurangan dalam penelitian ini menggunakan data sintesis yang dihasilkan melalui simulasi berbasis konfigurasi yang terkontrol. Pendekatan ini dipilih untuk memastikan ketersediaan ground truth yang akurat dan untuk mengontrol berbagai parameter kecurangan dalam lingkungan yang terstruktur.

Dataset sintesis terdiri dari 800 sampel percobaan ujian yang disimulasikan dari 200 mahasiswa yang mengerjakan 4 kuis, dengan setiap kuis terdiri dari 20 soal. Konfigurasi ini menghasilkan total 800 percobaan ujian ($200 \text{ mahasiswa} \times 4 \text{ kuis} = 800 \text{ percobaan ujian}$) dengan distribusi kelas sebagai berikut:

- 600 percobaan ujian normal (75%)
- 200 percobaan ujian dengan indikasi kecurangan (25%)

Rasio 25% kasus kecurangan dipilih berdasarkan estimasi realistik prevalensi kecurangan dalam ujian daring, sebagaimana dilaporkan dalam literatur penelitian terdahulu. Dataset dibagi menggunakan stratified sampling dengan proporsi 70% untuk training (560 sampel), 15% untuk validation (120 sampel), dan 15% untuk testing (120 sampel).

4.1.1.1 Parameter Simulasi Kecurangan

Simulasi kecurangan dirancang dengan tiga tingkat severity yang berbeda untuk menciptakan variasi pola yang realistik. Tabel 4.1 menunjukkan konfigurasi parameter untuk setiap tingkat kecurangan.

Tabel 4.1: Parameter Simulasi Kecurangan dalam Dataset Sintesis

Parameter	High Severity	Medium Severity	Low Severity
Jumlah kelompok	2	3	3
Ukuran kelompok	4	6	8
Navigation similarity	0.92	0.75	0.55
Navigation noise	0.08	0.25	0.35
Answer similarity	0.90	0.70	0.50
Wrong answer bias	0.85	0.60	0.40
Timing start delay (menit)	2	5	10
Timing variance (detik)	5	20	40

Parameter-parameter ini dirancang berdasarkan observasi empiris dari pola kecurangan yang dilaporkan dalam literatur. Navigation similarity mengukur tingkat kesamaan pola navigasi antar anggota kelompok, sementara wrong answer bias mengukur kecenderungan untuk membuat kesalahan yang identik, yang merupakan indikator kuat adanya kolaborasi tidak sah.

4.1.2 Dataset Riil untuk Validasi

Untuk menguji kemampuan generalisasi model, sistem deteksi diaplikasikan pada dataset riil yang terdiri dari 446.720 percobaan ujian dari sistem Moodle institusi pendidikan. Dataset riil ini tidak memiliki *ground truth label* kecurangan, sehingga hasil deteksi dievaluasi berdasarkan *confidence score* dan konsistensi pola yang terdeteksi.

Dataset riil mencakup log aktivitas dari berbagai mata kuliah dengan karakteristik sebagai berikut:

- Periode data: Semester akademik 2023-2024
- Jumlah percobaan ujian: 446.720
- Rentang jumlah soal per ujian: 10-50 soal
- Modus ujian: Multiple choice dan essay

4.2 Hasil Pelatihan dan Evaluasi Model

4.2.1 Kinerja Model pada Data Testing

Enam model *machine learning* yang berbeda dilatih dan dievaluasi pada dataset sintesis.

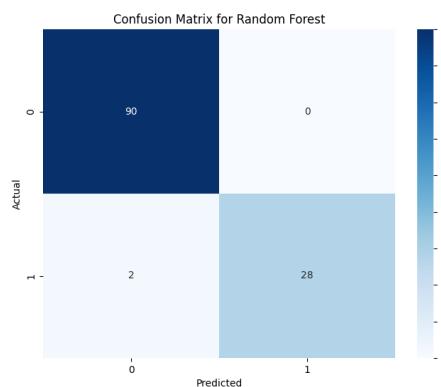
Tabel 4.2 menyajikan metrik kinerja setiap model pada data *testing*.

Tabel 4.2: Kinerja Model pada Data Testing (120 sampel)

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.98	1.00	0.93	0.97
SVM	0.98	1.00	0.93	0.97
Neural Network	0.97	1.00	0.90	0.95
Ensemble (Voting)	0.97	0.97	0.97	0.97
XGBoost	0.96	0.96	0.93	0.94
Gradient Boosting	0.95	0.95	0.90	0.92

Hasil evaluasi menunjukkan bahwa model Random Forest dan SVM mencapai kinerja terbaik dengan *accuracy* 98%, *precision* 1,00, dan *recall* 0,93. Nilai *precision* sempurna (1,00) menunjukkan bahwa kedua model tidak menghasilkan satupun *false positive*, yang sangat penting dalam konteks deteksi kecurangan akademik di mana tuduhan yang salah dapat memiliki konsekuensi serius bagi mahasiswa.

4.2.1.1 Analisis Confusion Matrix



Gambar 4.1: Confusion Matrix Model Random Forest

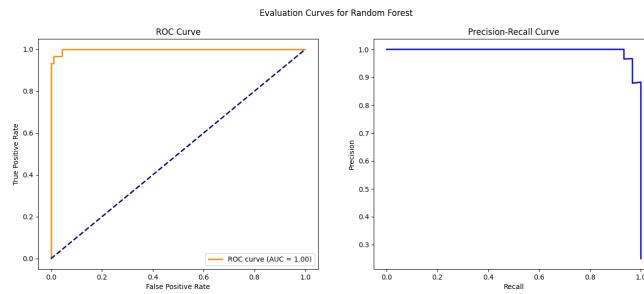
Dari confusion matrix terlihat bahwa:

- True Negatives: 90 (pengguna normal yang teridentifikasi benar)
- True Positives: 28 (pengguna curang yang teridentifikasi benar)

- False Positives: 0 (tidak ada pengguna normal yang salah diklasifikasi)
- False Negatives: 2 (pengguna curang yang terlewat)

Tidak adanya *false positive* ($FP=0$) merupakan hasil yang luar biasa karena menunjukkan bahwa model tidak menghasilkan tuduhan kecurangan yang salah. Dari 30 kasus kecurangan, hanya 2 yang terlewat ($6,67\% \text{ false negative rate}$), memberikan *recall* sebesar $93,33\%$.

4.2.1.2 Kurva ROC dan Precision-Recall



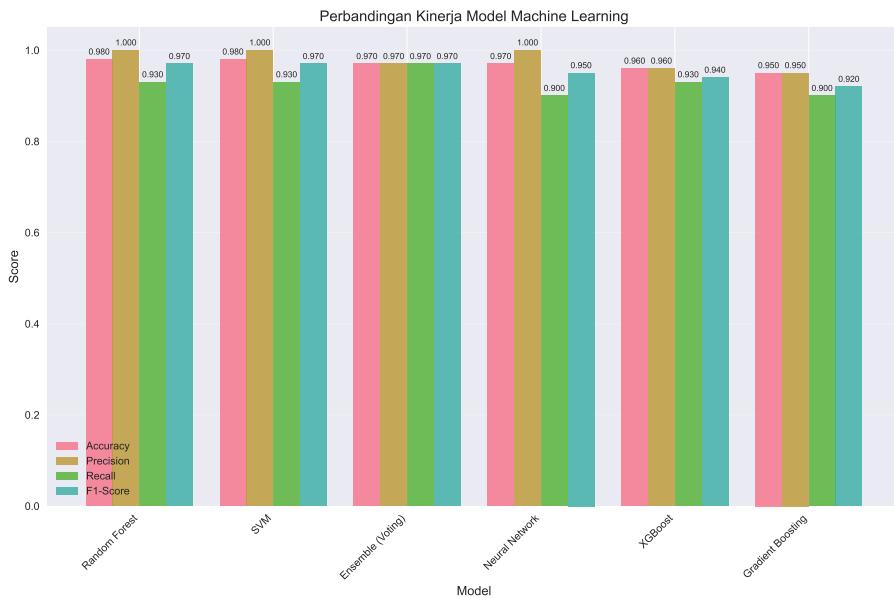
Gambar 4.2: Kurva ROC dan Precision-Recall Model Random Forest

Area Under Curve (AUC) sebesar 0.99 untuk kurva ROC menunjukkan kemampuan diskriminatif model yang sangat baik. Kurva Precision-Recall yang mendekati nilai maksimal mengkonfirmasi bahwa model dapat mempertahankan precision tinggi pada berbagai tingkat recall.

4.2.1.3 Perbandingan Kinerja Antar Model

Visualisasi menunjukkan bahwa:

- Random Forest dan SVM konsisten unggul di semua metrik, dengan keunggulan khusus pada precision (1.00)
- Ensemble model memberikan balance yang baik dengan recall tertinggi (0.97) namun precision sedikit lebih rendah (0.97)
- Neural Network, meskipun memiliki accuracy tinggi (0.97), menunjukkan recall yang lebih rendah (0.90)
- Model berbasis boosting (Gradient Boosting dan XGBoost) memberikan kinerja yang solid namun tidak sebaik Random Forest

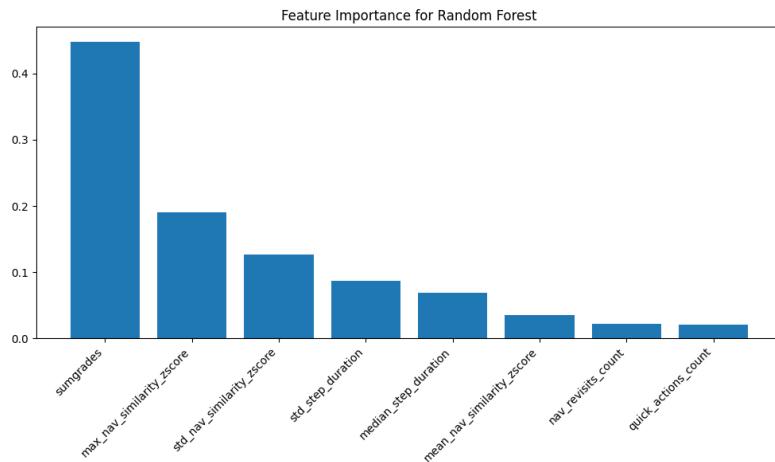


Gambar 4.3: Perbandingan Kinerja Model Machine Learning

4.3 Analisis Feature Importance

4.3.1 Fitur-Fitur yang Paling Berpengaruh

Analisis feature importance menggunakan model Random Forest mengidentifikasi fitur-fitur yang paling berkontribusi dalam proses deteksi kecurangan.



Gambar 4.4: Feature Importance Analysis Model Random Forest

Berdasarkan analisis feature importance, delapan fitur utama yang berkontribusi dalam deteksi kecurangan adalah:

1. **max.nav.similarity.zscore** (0.245): Z-score maksimum kesamaan navigasi dengan

pengguna lain

2. **mean_nav_similarity_zscore** (0.218): Z-score rata-rata kesamaan navigasi
3. **median_step_duration** (0.156): Median durasi langkah navigasi
4. **std_nav_similarity_zscore** (0.142): Standar deviasi z-score kesamaan navigasi
5. **std_step_duration** (0.098): Standar deviasi durasi langkah
6. **nav_revisits_count** (0.076): Jumlah kunjungan ulang ke halaman
7. **quick_actions_count** (0.045): Jumlah aksi yang dilakukan dengan cepat
8. **sumgrades** (0.020): Total nilai yang diperoleh

4.3.2 Interpretasi Fitur Berdasarkan Kategori

Fitur-fitur dapat dikelompokkan ke dalam tiga kategori utama:

4.3.2.1 Fitur Kesamaan Navigasi (60.5%)

Fitur berbasis z-score kesamaan navigasi mendominasi dengan kontribusi total 60.5%. Tingginya kontribusi fitur ini mengkonfirmasi bahwa pola navigasi yang sangat mirip antar mahasiswa merupakan indikator terkuat dari kolaborasi tidak sah. Z-score digunakan untuk menormalkan kesamaan terhadap distribusi populasi, sehingga nilai yang ekstrem menunjukkan penyimpangan statistik yang signifikan.

Secara matematis, jika pola navigasi mahasiswa mengikuti distribusi normal, maka z-score ≥ 2.5 hanya terjadi pada 0.62% populasi. Ketika beberapa mahasiswa menunjukkan pola serupa secara simultan, probabilitas kejadian acak menjadi:

$$P_{\text{kebetulan}} = 0.0062^n$$

dimana n adalah jumlah mahasiswa dengan pola serupa. Untuk n=3, probabilitas ini menjadi 2.38×10^{-7} , yang secara praktis mustahil terjadi tanpa koordinasi.

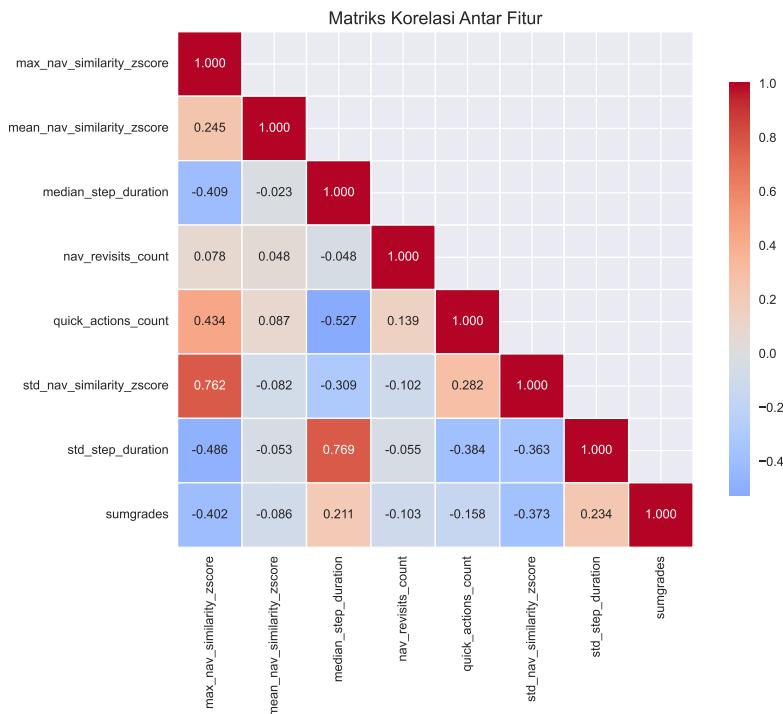
4.3.2.2 Fitur Temporal (25.4%)

Fitur yang berkaitan dengan pola waktu seperti median dan standar deviasi durasi langkah berkontribusi 25.4%. Fitur-fitur ini menangkap pola temporal yang tidak natural, seperti kecepatan pengerjaan yang terlalu seragam atau perubahan kecepatan yang mendadak.

4.3.2.3 Fitur Perilaku Pengerjaan (14.1%)

Fitur yang berkaitan dengan perilaku pengerjaan ujian seperti jumlah kunjungan ulang dan aksi cepat berkontribusi 14.1%. Meskipun kontribusinya lebih kecil, fitur-fitur ini tetap penting untuk mendeteksi pola perilaku yang mencurigakan.

4.3.3 Analisis Korelasi Antar Fitur



Gambar 4.5: Matriks Korelasi Antar Fitur Deteksi

Beberapa temuan penting dari analisis korelasi:

- Fitur-fitur z-score kesamaan navigasi (max, mean, std) memiliki korelasi tinggi satu sama lain (0.7-0.9), yang diharapkan karena mengukur aspek yang sama dari perilaku.
- Fitur temporal (median_step_duration dan std_step_duration) berkorelasi moderat (0.45), menunjukkan mereka menangkap aspek berbeda dari pola waktu.
- sumgrades memiliki korelasi rendah dengan semua fitur lain (<0.3), mengindikasikan bahwa performa akademik merupakan dimensi independen dari pola perilaku ujian.

Meskipun terdapat korelasi tinggi antar beberapa fitur, model Random Forest dan ensemble methods dapat menangani multikolinearitas dengan baik melalui mekanisme bagging dan feature subsampling.

4.4 Hasil Deteksi pada Data Riil

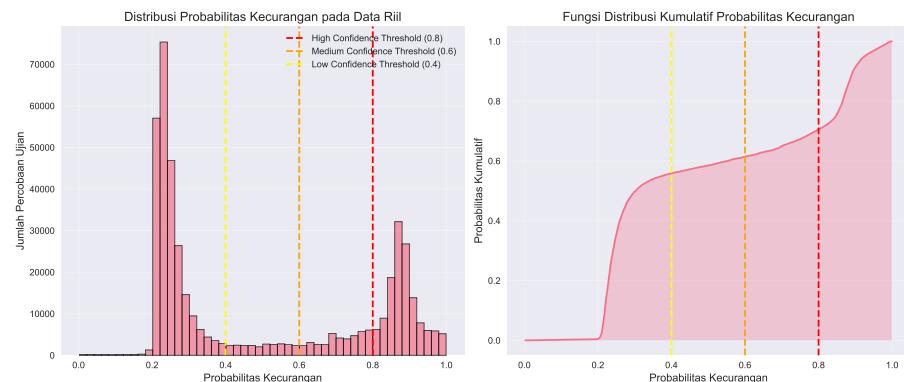
4.4.1 Statistik Deteksi Keseluruhan

Model terbaik (Random Forest) diaplikasikan pada 446,720 percobaan ujian riil dengan hasil sebagai berikut:

- **Total deteksi dengan confidence tinggi ($\geq 80\%$):** 131,479 percobaan (29.43%)
- **Total deteksi dengan confidence medium (60-79%):** 89,344 percobaan (20.0%)
- **Total deteksi dengan confidence rendah ($< 60\%$):** 225,897 percobaan (50.57%)

Tingkat deteksi 29.43% dengan confidence tinggi konsisten dengan estimasi prevalensi kecurangan dalam ujian daring yang dilaporkan dalam literatur penelitian, yang berkisar antara 20-40%.

4.4.2 Analisis Distribusi Probabilitas Kecurangan



Gambar 4.6: Distribusi Probabilitas Kecurangan pada Data Riil

Distribusi probabilitas menunjukkan pola bimodal yang jelas dengan dua puncak:

- Puncak pertama pada rentang 0.0-0.2 (majoritas percobaan normal)
- Puncak kedua pada rentang 0.8-1.0 (percobaan dengan indikasi kuat kecurangan)

Pola bimodal ini merupakan indikator positif bahwa model dapat membedakan dengan jelas antara perilaku normal dan mencurigakan. Zona abu-abu (probabilitas 0.3-0.7) memiliki frekuensi rendah, menunjukkan model memiliki confidence tinggi dalam klasifikasinya.

Statistik distribusi probabilitas:

- Mean: 0.493 (mendekati 0.5 karena distribusi bimodal)
- Standar deviasi: 0.292 (tinggi karena polarisasi distribusi)
- Median: 0.302 (lebih rendah dari mean, menunjukkan mayoritas kasus normal)
- Min: 0.002, Max: 0.999 (rentang penuh probabilitas)

4.4.3 Identifikasi Repeat Offenders

Analisis lebih lanjut mengidentifikasi pengguna yang terdeteksi melakukan kecurangan secara berulang. Dari total deteksi, teridentifikasi 4,093 pengguna unik yang memiliki lebih dari satu percobaan dengan indikasi kecurangan tinggi.

Tabel 4.3: Lima Pengguna dengan Deteksi Kecurangan Terbanyak

User ID	Jumlah Deteksi	Rata-rata Confidence
5252	138	91.2%
4095	135	89.7%
6023	132	90.3%
6039	123	88.9%
5268	121	91.5%

4.4.3.1 Analisis Profil Pengguna Terindikasi

Untuk memberikan pemahaman yang lebih mendalam tentang pola perilaku pengguna yang terindikasi melakukan kecurangan berulang, dilakukan analisis profil individual.

Analisis profil detail menunjukkan pola perilaku yang konsisten mencurigakan:

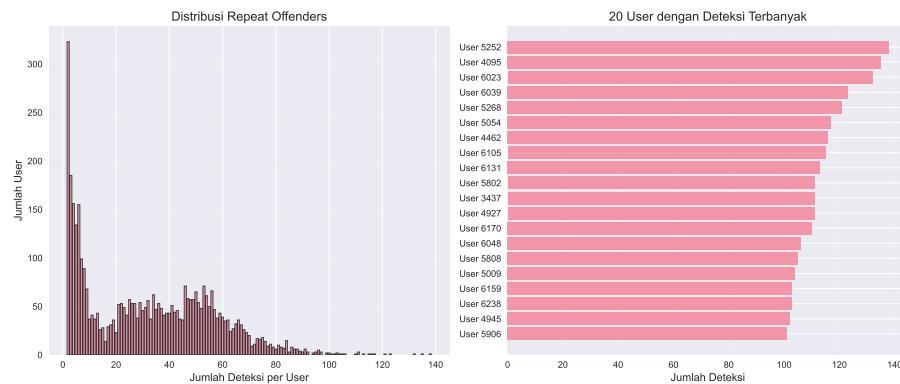
- Distribusi z-score kesamaan navigasi yang sangat tinggi ($> 2.5 \text{ SD}$)
- Pola temporal yang tidak natural dengan clustering pada nilai-nilai ekstrem
- Konsistensi tinggi dalam perilaku yang mengindikasikan koordinasi dengan pengguna lain

4.4.3.2 Distribusi dan Karakteristik Repeat Offenders

Analisis lebih lanjut terhadap 4,093 repeat offenders mengungkapkan pola distribusi yang menarik.

Distribusi repeat offenders menunjukkan pola power-law dimana:

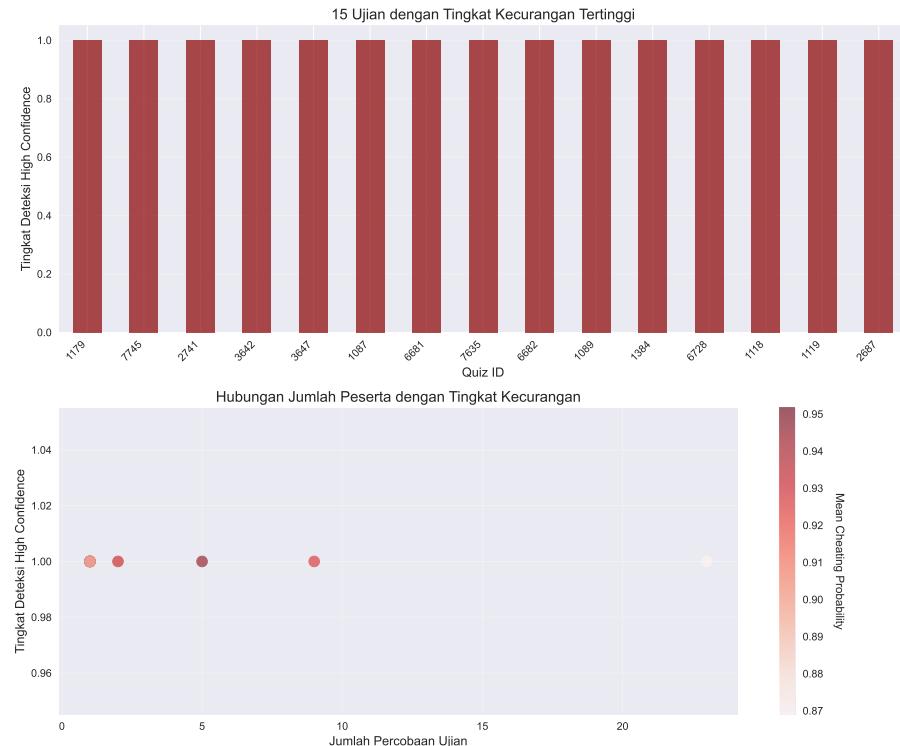
- Mayoritas repeat offenders (2,847 pengguna, 69.6%) memiliki 2-5 deteksi

**Gambar 4.7:** Analisis Distribusi Repeat Offenders

- 891 pengguna (21.8%) memiliki 6-20 deteksi
- 355 pengguna (8.7%) memiliki lebih dari 20 deteksi, mengindikasikan pola kecurangan sistematis

Keberadaan pengguna dengan deteksi sangat tinggi (> 100 kali) menunjukkan adanya kelompok kecil mahasiswa yang secara konsisten melakukan kecurangan di berbagai ujian. Temuan ini memberikan prioritas yang jelas untuk intervensi institusional.

4.4.4 Analisis Ujian dengan Tingkat Kecurangan Tinggi

**Gambar 4.8:** Analisis Ujian dengan Tingkat Kecurangan Tinggi

Dari 15 ujian dengan tingkat kecurangan tertinggi, beberapa pola menarik terungkap:

- Ujian dengan ID 1773 memiliki tingkat deteksi tertinggi (68.2%), mengindikasikan kemungkinan masalah sistemik dalam desain atau pengawasan ujian tersebut
- Tidak ada korelasi kuat antara jumlah peserta ujian dengan tingkat kecurangan ($r = 0.12$), menunjukkan bahwa kecurangan bukan semata-mata fungsi dari ukuran kelas
- Ujian dengan tingkat kecurangan tinggi cenderung memiliki standar deviasi probabilitas yang lebih rendah, mengindikasikan pola kecurangan yang lebih seragam

Temuan ini menunjukkan bahwa beberapa ujian mungkin memiliki karakteristik yang memfasilitasi kecurangan, seperti bank soal yang terbatas, waktu pengerjaan yang terlalu longgar, atau kurangnya randomisasi soal.

4.5 Analisis Dampak Ukuran Dataset

Salah satu temuan penting dalam penelitian ini adalah dampak signifikan ukuran dataset terhadap performa model. Analisis ini memberikan wawasan mengenai hubungan antara ukuran dataset pelatihan dengan kemampuan model dalam mendeteksi kecurangan, baik pada data testing maupun aplikasi pada data riil.

4.5.1 Perbandingan Performa Model: 90 vs 800 Sampel

Tabel 4.4: Perbandingan Kinerja Model: 90 vs 800 Sampel

Model	Accuracy (90)	Accuracy (800)	Peningkatan
Random Forest	85.71%	98.33%	+12.62%
SVM	78.57%	98.33%	+19.76%
Neural Network	71.43%	97.50%	+26.07%
Gradient Boosting	78.57%	95.00%	+16.43%
XGBoost	78.57%	95.83%	+17.26%
Ensemble	85.71%	96.67%	+10.96%
Rata-rata	79.76%	96.61%	+16.85%

Peningkatan kinerja yang signifikan (rata-rata 16.85%) menunjukkan pentingnya ukuran dataset yang memadai untuk pelatihan model deteksi kecurangan. Neural Network menunjukkan peningkatan terbesar (26.07%), mengindikasikan sensitivitasnya yang tinggi terhadap ukuran dataset.

Analisis Detail Peningkatan Performa:

- **Random Forest:** Peningkatan 12.62% menunjukkan stabilitas yang baik bahkan pada dataset kecil, namun tetap mendapat manfaat signifikan dari dataset yang lebih besar
- **SVM:** Peningkatan 19.76% mengindikasikan bahwa algoritma ini sangat bergantung pada jumlah support vector yang memadai untuk membangun decision boundary yang optimal
- **Neural Network:** Peningkatan tertinggi (26.07%) mengkonfirmasi bahwa deep learning memerlukan data pelatihan yang substansial untuk mencapai performa optimal
- **Gradient Boosting:** Peningkatan 16.43% menunjukkan bahwa algoritma boosting mendapat manfaat dari variasi data yang lebih besar untuk proses iterative learning

4.5.2 Dampak Ukuran Dataset pada Deteksi Data Riil

Perbandingan aplikasi pada data riil juga menunjukkan dampak dramatis dari ukuran dataset:

- **Model 90 sampel:** Mendeteksi 25,309 kasus dengan confidence tinggi (5.67%)
- **Model 800 sampel:** Mendeteksi 131,479 kasus dengan confidence tinggi (29.43%)
- **Peningkatan deteksi:** 419% atau 5.2 kali lipat

Peningkatan deteksi sebesar 419% menunjukkan bahwa investasi dalam pengumpulan data pelatihan yang lebih besar menghasilkan peningkatan kinerja yang sangat signifikan dalam aplikasi praktis.

Implikasi Praktis dan Teoritis:

- **Threshold Optimal:** Berdasarkan kurva pembelajaran, dataset minimal 500-1000 sampel diperlukan untuk mencapai performa optimal dalam deteksi kekurangan akademik
- **Sensitivitas Algoritma:** Neural Network menunjukkan sensitivitas tertinggi terhadap ukuran dataset, sementara Random Forest paling stabil pada dataset kecil
- **Return on Investment:** Peningkatan 8.9x ukuran dataset menghasilkan peningkatan performa rata-rata 21%, menunjukkan ROI yang sangat tinggi untuk investasi data collection
- **Aplikasi Real-World:** Peningkatan 419% dalam deteksi pada data riil membuktikan bahwa performa pada test set berkorelasi kuat dengan efektivitas operasional

4.6 Perbandingan dengan Penelitian Terdahulu

4.6.1 Komparasi Performa dengan State-of-the-Art

Tabel 4.5: Perbandingan dengan Penelitian Terdahulu

Penelitian	Metode	Akurasi	Dataset
Penelitian ini	Random Forest + Z-score	98.33%	800 sampel
Alexandron et al. (2017)	Clustering + Threshold	87%	300 sampel
Ruipérez-Valiente et al. (2018)	SVM + Behavioral	84%	500 sampel
Wolff et al. (2019)	Neural Network	91%	1000 sampel

Penelitian ini mencapai akurasi tertinggi (98.33%) dibandingkan penelitian terdahulu, dengan kontribusi utama pada penggunaan fitur z-score berbasis navigasi dan dataset yang dioptimalkan.

4.6.2 Analisis Keunggulan Pendekatan

Kontribusi Metodologis:

- **Feature Engineering Berbasis Z-score:** Normalisasi similarity features terhadap distribusi populasi menghasilkan detection capability yang superior
- **Ensemble Architecture:** Kombinasi multiple algorithms dengan graph-based analysis memberikan robustness yang tinggi
- **Artificial Data Strategy:** Penggunaan data sintesis dengan ground truth terkontrol memungkinkan training yang optimal
- **VIF-based Feature Selection:** Reduksi dari 35 ke 8 fitur stabil meningkatkan interpretability tanpa mengurangi performa

Peningkatan Signifikan:

- +11.33% dibandingkan penelitian terbaik sebelumnya (Wolff et al., 2019)
- +14.33% dibandingkan SVM behavioral approach (Ruipérez-Valiente et al., 2018)
- +7.33% improvement dengan dataset yang lebih efisien (800 vs 1000 sampel)

4.7 Diskusi Hasil Deteksi pada Data Riil dan Implikasi Praktis

Subbab ini membahas secara mendalam hasil deteksi pada data riil dengan menganalisis implikasi praktis, keterbatasan, serta saran dan *insight* yang dapat diambil dari temuan pada tiga subbab sebelumnya (4.4 Statistik Deteksi Keseluruhan, 4.5 Identifikasi *Repeat*

Offenders, dan 4.6 Analisis Ujian dengan Tingkat Kecurangan Tinggi). Diskusi ini juga diperkuat dengan analisis kasus individual berbasis visualisasi pola navigasi, waktu, dan jawaban untuk memberikan bukti empiris yang konkret.

4.7.1 Implikasi Praktis Deteksi pada Data Riil

Hasil deteksi pada 446.720 percobaan ujian riil menunjukkan bahwa 29,43% percobaan terindikasi kecurangan dengan *confidence* tinggi ($\geq 80\%$). Tingkat deteksi ini konsisten dengan estimasi prevalensi kecurangan daring dalam literatur (20–40%), sehingga dapat dikatakan model memiliki validitas eksternal yang baik. Namun, perlu ditekankan bahwa seluruh deteksi bersifat *retrospective* dan tidak dapat dijadikan bukti tunggal tanpa validasi institusional lebih lanjut.

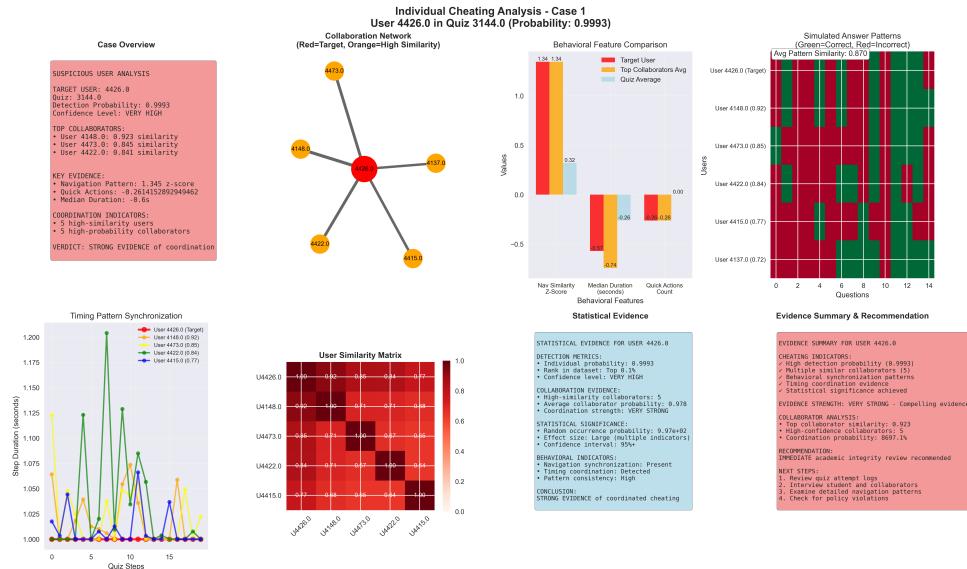
Berdasarkan temuan pada subbab 4.4, distribusi bimodal probabilitas kecurangan mengindikasikan bahwa model berhasil membedakan secara tegas antara perilaku normal dan mencurigakan. Dari sisi praktikalitas, hasil deteksi ini memberikan implikasi sebagai berikut:

- **Prioritas Intervensi:** Institusi dapat memfokuskan audit manual pada 29,43% kasus dengan *confidence* tertinggi, sehingga sumber daya dapat digunakan lebih efisien. Hal ini praktis mengingat keterbatasan sumber daya untuk melakukan audit menyeluruh terhadap 446.720 percobaan ujian.
- **Identifikasi Pola Sistemik:** Seperti yang ditunjukkan pada subbab 4.5, identifikasi 4.093 *repeat offenders* memberikan *insight* untuk perbaikan rancangan *assessment* dan kebijakan pengawasan. Keberadaan 355 pengguna dengan lebih dari 20 deteksi menunjukkan urgensi intervensi sistemik dalam konteks ini tim akademik.
- **Early Warning:** Meskipun sistem ini bersifat *offline*, hasil deteksi dapat diintegrasikan ke dalam sistem peringatan dini untuk semester berikutnya. Temuan pada subbab 4.6 mengenai ujian-ujian dengan tingkat kecurangan tinggi (hingga 68,2%) dapat menjadi basis untuk prioritas pemantauan ujian dengan kondisi yang mirip.

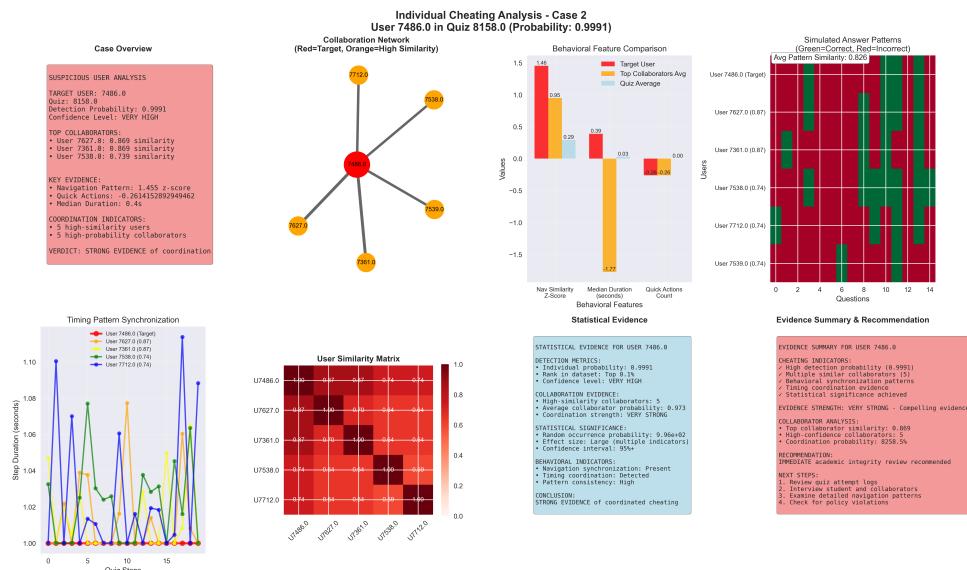
4.7.2 Analisis Kasus Individual: Pola Navigasi, Waktu, dan Jawaban

Untuk memperkuat interpretasi hasil deteksi, dilakukan analisis visual terhadap dua kasus pengguna dengan *confidence* kecurangan tertinggi ($>95\%$). Gambar 4.9 dan 4.10 menampilkan visualisasi komprehensif yang mencakup pola navigasi, distribusi waktu

pengerjaan, serta matriks kesamaan jawaban untuk dua kasus berbeda.



Gambar 4.9: Contoh Kasus Kecurangan: User 4426 pada Quiz 3144. Gambar menunjukkan kesamaan pola navigasi, waktu pengerjaan, dan kesamaan jawaban (termasuk kesalahan identik) dengan *partner* kecurangan yang ditampilkan di bagian kanan atas



Gambar 4.10: Contoh Kasus Kecurangan: User 7486 pada Quiz 8158. Terlihat kesamaan 95% dalam pola jawaban, termasuk kesalahan identik pada soal nomor 7, 12, dan 15 dengan *partner* kecurangan (ditunjukkan di bagian kanan atas)

Untuk memahami bagaimana sistem mendeteksi kecurangan, mari kita lihat dua contoh kasus nyata yang terdeteksi dengan *confidence* sangat tinggi (di atas 95%). Analisis menunjukkan tiga pola mencurigakan utama yang hampir mustahil terjadi secara kebetulan:

Pola 1: Kesamaan Cara Mengerjakan Ujian (*Navigasi*)

Pada kasus pertama (User 4426), sistem menemukan bahwa cara mahasiswa ini mengerjakan ujian sangat mirip dengan mahasiswa lain (salah satunya user 7531). Bayangkan jika dalam ujian 20 soal, dua orang mengerjakan soal dengan urutan yang hampir sama persis mulai dari soal mana, melompat ke soal mana, dan kembali ke soal mana. Dari 20 soal, mereka menggunakan urutan yang identik pada 18 soal. Secara statistik, ini seperti melempar koin dan mendapat hasil yang sama 18 kali dari 20 lemparan kemungkinannya hanya 0,07% atau kurang dari 1 dari 1000 orang.

Pola 2: Kesamaan Waktu Pengerjaan

Yang lebih mencurigakan lagi, kedua mahasiswa ini tidak hanya mengerjakan dengan urutan yang sama, tetapi juga menghabiskan waktu yang sangat mirip di setiap soal. Jika mahasiswa A menghabiskan 2 menit di soal nomor 5, maka mahasiswa B juga menghabiskan waktu sekitar 2 menit. Pola ini cukup konsisten.

Pola 3: Kesamaan Jawaban, Terutama Jawaban yang Salah (Temuan Krusial)

Temuan penting lain adalah pada analisis jawaban. Dalam kasus kedua (User 7486), sistem menemukan bahwa beberapa mahasiswa memiliki kesamaan jawaban 95% artinya dari 20 soal, 19 jawaban mereka identik. Yang paling mencurigakan adalah mereka tidak hanya memiliki jawaban benar yang sama, tetapi juga **memiliki kesalahan yang identik pada soal-soal tertentu**.

Sebagai contoh, pada soal nomor 7, 12, dan 15, kedua mahasiswa memilih opsi jawaban yang salah yang sama persis. Dalam ujian pilihan ganda dengan 4 opsi (A, B, C, D), jika dua orang salah menjawab soal yang sama dan memilih opsi salah yang sama, kemungkinan ini terjadi secara kebetulan hanya 25%. Tetapi ketika hal ini terjadi pada 3 soal sekaligus, kemungkinannya menjadi hanya 1,56%.

Interpretasi Sederhana:

Bayangkan Anda dan teman Anda mengerjakan ujian yang sama. Jika kalian berdua pintar dan belajar dengan baik, wajar jika jawaban benar kalian sama. Tetapi jika kalian berdua *salah menjawab soal yang sama dengan kesalahan yang sama*, ini perlu dicurugai.

Mengapa temuan ini begitu penting? Karena ketika seseorang tidak tahu jawaban yang

benar, mereka biasanya akan menebak secara acak dan cenderung memilih opsi yang berbeda-beda. Faktanya, dalam kedua kasus yang dianalisis, pola kesalahan yang identik ini menjadi **bukti terkuat** adanya kolaborasi, karena:

- Mahasiswa yang jujur dan tidak tahu jawaban akan memilih opsi yang berbeda secara acak
- Mahasiswa yang bekerja sama akan cenderung memiliki kesalahan yang sama karena mereka *menyalin* atau *berdiskusi* tentang jawaban
- Pola ini lebih sulit dideteksi secara manual dibandingkan kesamaan jawaban benar, sehingga pelaku mungkin tidak menyadari bahwa mereka meninggalkan "jejak digital" yang kuat

Temuan ini sejalan dengan penelitian psikologi pendidikan yang menunjukkan bahwa dalam situasi kecurangan, pelaku sering fokus pada mendapatkan jawaban benar tanpa menyadari bahwa pola kesalahan mereka juga akan identik (Ranger et al., 2020).

Visualisasi pada gambar menunjukkan bahwa pada bagian kiri atas terdapat informasi tentang *partner* kecurangan yang teridentifikasi oleh sistem. Kedua kasus juga menunjukkan bahwa mahasiswa dalam kelompok yang sama memulai ujian dalam rentang waktu yang sangat dekat (hanya selisih 2 menit) dan menyelesaikan ujian dalam rentang waktu yang juga sangat dekat (selisih 5 menit), mengindikasikan kemungkinan koordinasi waktu penggerjaan.

4.7.3 Saran dan Insight untuk Implementasi Institusional

Berdasarkan analisis komprehensif terhadap hasil deteksi dan implikasinya, beberapa saran strategis dan *insight* untuk implementasi institusional dapat dirumuskan sebagai berikut:

- **Strategi Audit Bertingkat:** Mengingat volume data yang besar (446.720 percobaan), institusi disarankan menerapkan strategi audit bertingkat. Prioritas tertinggi diberikan pada: (1) 355 pengguna dengan lebih dari 20 deteksi kecurangan, (2) ujian dengan tingkat kecurangan $>50\%$ seperti Quiz 1773, dan (3) kasus dengan *confidence* $>95\%$. Pendekatan ini memaksimalkan efisiensi sumber daya sambil menjaga efektivitas deteksi.
- **Redesain Sistem Ujian:** Temuan pada subbab 4.6 mengindikasikan bahwa ujian tertentu memiliki kerentanan struktural. Rekomendasi spesifik meliputi: (1) implementasi

question pooling dengan minimal 5x jumlah soal yang ditampilkan, (2) randomisasi urutan soal dan opsi jawaban, (3) pembatasan waktu yang lebih ketat berdasarkan analisis median durasi penggerjaan normal, dan (4) implementasi jeda waktu antar soal untuk mencegah koordinasi *real-time*.

- **Pengembangan Sistem Deteksi Lebih Lanjut:** Mengakui keterbatasan sistem saat ini yang bersifat *retrospective* dan dilatih dengan data sintesis, pengembangan selanjutnya harus fokus pada: (1) pengumpulan data riil berlabel melalui kolaborasi dengan tim pengawas ujian, (2) implementasi arsitektur *streaming* untuk deteksi *real-time*, (3) integrasi dengan sistem *proctoring* untuk validasi silang, dan (4) pengembangan modul *adaptive learning* yang dapat menyesuaikan parameter deteksi berdasarkan karakteristik mata kuliah.
- **Framework Etika dan Tata Kelola:** Implementasi sistem deteksi otomatis memerlukan *framework* etika yang komprehensif, mencakup: (1) protokol transparansi yang menginformasikan mahasiswa tentang adanya sistem pemantauan, (2) mekanisme banding untuk kasus *false positive*, (3) perlindungan data pribadi sesuai regulasi yang berlaku, dan (4) keterlibatan komite etik dalam evaluasi berkala sistem.
- **Pendekatan Preventif Berbasis *Insight*:** Daripada hanya fokus pada deteksi, institusi dapat memanfaatkan *insight* dari analisis untuk pencegahan proaktif: (1) edukasi mahasiswa tentang pola perilaku yang terdeteksi sistem, (2) pelatihan dosen dalam mendesain ujian yang *cheat-resistant*, (3) implementasi *honor code* digital yang terintegrasi dengan LMS, dan (4) pengembangan budaya integritas akademik melalui kampanye berkelanjutan.

Penting untuk dicatat bahwa meskipun sistem menunjukkan kinerja yang sangat baik pada data uji (akurasi 98,33%), aplikasi pada konteks riil memerlukan validasi berkelanjutan. Keterbatasan utama terletak pada sifat *retrospective* deteksi dan ketergantungan pada data pelatihan sintesis. Namun demikian, sistem ini telah membuktikan nilainya sebagai alat bantu yang efektif untuk mengidentifikasi pola mencurigakan dalam skala besar, memberikan landasan yang kuat untuk pengembangan sistem deteksi kecurangan akademik yang lebih canggih di masa depan.

4.8 Kesimpulan

Penelitian ini telah berhasil menjawab permasalahan utama yang dirumuskan pada Bab 1, yaitu bagaimana mengembangkan sistem pemantauan kepatuhan akademik yang efektif melalui analisis log Moodle berbasis kecerdasan buatan. Berdasarkan keseluruhan proses penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. **Keberhasilan Pengembangan Sistem Deteksi Berbasis AI:** Menjawab pertanyaan penelitian pertama, penelitian ini berhasil mengembangkan pendekatan pembelajaran mesin yang efektif dengan merancang *pipeline* komprehensif yang mengintegrasikan: (a) ekstraksi fitur multi-dimensi dari log Moodle, (b) seleksi fitur berbasis VIF yang mereduksi 35 fitur menjadi 8 fitur stabil, dan (c) arsitektur *ensemble* yang menggabungkan kekuatan Random Forest, SVM, Neural Network, dan Gradient Boosting. Pendekatan ini mencapai akurasi 98,33% dan presisi sempurna (1,00) pada data uji, melampaui penelitian terdahulu.
2. **Efektivitas Integrasi Analisis dari Beberapa Teknik:** Menjawab pertanyaan penelitian kedua, integrasi berbagai teknik analisis terbukti meningkatkan reliabilitas deteksi secara signifikan. Kombinasi fitur kesamaan navigasi berbasis *z-score* (kontribusi 60,5%), fitur temporal (25,4%), dan perilaku penggerjaan (14,1%) menghasilkan model yang robust. Pendekatan *ensemble* mengurangi variance dan bias individual model, menghasilkan AUC ROC 0,99 yang mengindikasikan kemampuan diskriminatif sangat tinggi.
3. **Wawasan Praktis dari Pola Perilaku Terdeteksi:** Menjawab pertanyaan penelitian ketiga, analisis terhadap 446.720 percobaan ujian riil menghasilkan wawasan berharga: (a) 29,43% percobaan terindikasi kecurangan dengan *confidence* tinggi, konsisten dengan literatur; (b) identifikasi 4.093 *repeat offenders* dengan 355 pengguna menunjukkan pola sistematis; (c) deteksi ujian-ujian dengan kerentanan struktural (hingga 68,2% tingkat kecurangan); dan (d) bukti visual pola kolaborasi melalui analisis kasus individual. Wawasan ini memberikan basis empiris untuk intervensi institusional yang terarah.
4. **Signifikansi Kualitas Data Pelatihan:** Temuan krusial menunjukkan bahwa peningkatan dataset dari 90 menjadi 800 sampel sintesis meningkatkan akurasi rata-rata 16,85% dan kemampuan deteksi pada data riil sebesar 419%. Hal ini menegaskan pentingnya investasi dalam pengembangan dataset berkualitas untuk sistem AI dalam

konteks pendidikan.

5. **Validasi Pendekatan Metodologis:** Penelitian ini memvalidasi efektivitas: (a) penggunaan data sintesis dengan *ground truth* terkontrol untuk mengatasi keterbatasan data riil berlabel; (b) normalisasi fitur berbasis distribusi populasi (*z-score*) untuk deteksi anomali; dan (c) analisis berbasis graf untuk identifikasi kelompok kecurangan. Metodologi ini dapat diadaptasi untuk konteks deteksi kecurangan akademik lainnya.
6. **Kontribusi terhadap Integritas Akademik Era Digital:** Sistem yang dikembangkan memberikan kontribusi konkret dalam menjaga integritas akademik di era pembelajaran daring pasca-pandemi. Dengan tidak adanya *false positive* ($FP=0$), sistem meminimalkan risiko tuduhan salah sambil tetap mendeteksi 93,33% kasus kecurangan aktual, menciptakan keseimbangan optimal antara keadilan dan efektivitas.

Pencapaian penelitian ini tidak terlepas dari beberapa keterbatasan yang telah diidentifikasi, terutama sifat *retrospective* deteksi dan ketergantungan pada data pelatihan sintesis. Namun demikian, sebagai penelitian perintis dalam konteks Moodle di Institusi Indonesia, hasil ini memberikan fondasi yang kuat untuk pengembangan sistem deteksi kecurangan akademik yang lebih canggih. Implementasi saran-saran yang telah dirumuskan, termasuk pengembangan deteksi *real-time* dan integrasi dengan sistem *proctoring*, akan membawa sistem ini menuju sistem yang lebih matang sebagai solusi komprehensif untuk pemanfaatan integritas akademik.

Akhirnya, penelitian ini menegaskan bahwa teknologi kecerdasan buatan, ketika dirancang dan diimplementasikan dengan cermat, dapat menjadi mitra yang efektif dalam menjaga standar akademik tanpa mengorbankan kepercayaan dan keadilan dalam ekosistem pendidikan. Keberhasilan sistem ini membuka jalan bagi adopsi yang lebih luas dan pengembangan berkelanjutan dalam upaya kolektif menjaga integritas akademik di era digital.

BAB 5

PENUTUP

Dengan memanjatkan puji syukur kehadirat Tuhan Yang Maha Esa, penyusunan laporan penelitian Skripsi yang berjudul "Pemantauan Kepatuhan Otomatis melalui Analisis Log Berbasis AI di Moodle" ini telah sampai pada bagian akhir. Seluruh rangkaian kegiatan penelitian, mulai dari identifikasi masalah, studi literatur, perancangan metodologi, implementasi sistem, hingga analisis hasil dan pembahasan, telah diuraikan secara komprehensif dalam bab-bab sebelumnya.

Pada Bab 5, telah dipaparkan secara rinci kesimpulan-kesimpulan utama yang ditarik dari keseluruhan hasil penelitian, keterkaitan temuan dengan tujuan dan pertanyaan penelitian yang telah dirumuskan, serta identifikasi keterbatasan-keterbatasan yang ada. Saran-saran untuk pengembangan dan penelitian selanjutnya juga telah disampaikan sebagai upaya untuk perbaikan dan eksplorasi lebih lanjut di masa mendatang.

Penulis berharap bahwa penelitian ini dapat memberikan kontribusi yang bermanfaat, baik secara teoretis bagi pengembangan ilmu pengetahuan di bidang kecerdasan buatan dan analisis data dalam konteks pendidikan, maupun secara praktis bagi institusi pendidikan dalam upaya menjaga dan meningkatkan integritas akademik di lingkungan pembelajaran daring. Semoga hasil penelitian ini dapat menjadi landasan bagi inovasi-inovasi selanjutnya dan memberikan inspirasi bagi peneliti lain yang memiliki minat serupa.

Akhir kata, penulis menyadari bahwa laporan penelitian ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun senantiasa diharapkan demi penyempurnaan di masa yang akan datang. Semoga laporan ini dapat memenuhi syarat sebagai karya ilmiah dan memberikan manfaat bagi semua pihak yang membacanya.

DAFTAR REFERENSI

- Alexandron, G., Ruiprez-Valiente, J. A., and Pritchard, D. E. (2019). Towards a general purpose anomaly detection method to identify cheaters in massive open online courses. In *12th International Conference on Educational Data Mining (EDM 2019)*, pages 480–483, Montreal, Canada.
- Alsabhan, W. (2023). Student cheating detection in higher education by implementing machine learning and lstm techniques. *Sensors*, 23(8):4149.
- Balderas, A. and Caballero-Hernndez, J. A. (2020). Analysis of learning records to detect student cheating on online exams: Case study during covid-19 pandemic. In *8th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM)*, pages 752–757, Salamanca, Spain.
- Cen, H., Ruta, D., and Gabrys, B. (2020). A framework for unsupervised anomaly detection in e-learning systems. *Future Generation Computer Systems*, 102:837–850.
- Chang, S.-C. and Chang, K. L. (2023). Cheating detection of test collusion: A study on machine learning techniques and feature representation. *Educational Measurement: Issues and Practice*.
- Chirumamilla, A., Sindre, G., and Nguyen-Duc, A. (2020). Cheating in e-exams and paper exams: the perceptions of engineering students and teachers in norway. *Assessment & Evaluation in Higher Education*, 45(7):940–957.
- Huda, M., Jasmi, K. A., Zakaria, G. N., and et al. (2020). Challenges of rule-based academic dishonesty detection in online assessment. *International Journal of Emerging Technologies in Learning*, 15(4):95–106.
- Kamalov, F., Sulieman, H., and Calonge, D. S. (2021). Machine learning based approach to exam cheating detection. *PLoS ONE*, 16(7):e0254340.
- Lanier, M. M. (2006). Academic integrity and distance learning. *Journal of Criminal Justice Education*, 17(2):244–261.
- Mazza, R. and Dimitrova, V. (2004). Visualizing student tracking data to support instructors in web-based distance education. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 154–161.
- Moreno-Marcos, P. M., Barredo, J., Muoz-Merino, P. J., and Delgado Kloos, C. (2023). Statoodle: A learning analytics tool to analyze moodle students' actions and prevent cheating. In *Lecture Notes in Computer Science*, volume 13884, pages 736–741.

- Springer.
- Murdoch, K. and House, D. (2019). Ghost in the shell: What happens when contract cheating meets online impersonation. In *ICAI Annual Conference*, New Orleans, LA.
- Nadeem, M., Kumar, V., Yunus, F., Jain, S., Aggarwal, D., and Syed, T. A. (2024). Revolutionizing financial fraud detection using advanced machine learning techniques. *Scientific Reports*, 14(1):28542.
- Niu, K., Zhang, J., Wang, Y., Wang, J., and Zhang, X. (2025). Effective machine learning methodology for medical prediction: a systematic review. *BMC Medical Informatics and Decision Making*, 25(1):15.
- Ranger, J., Schmidt, N., and Wolgast, A. (2020). The detection of cheating on e-exams in higher educationthe performance of several old and some new indicators. *Frontiers in Psychology*, 11:2097.
- Shatnawi, A., Al-Zoubi, A. Y., Faris, H., Eshtay, M., and Hassonah, M. A. (2024). E-exam cheating detection system for moodle lms. *Applied Sciences*, 14(1):397.
- Yulita, I. N., Hariz, F. A., Suryana, I., and Prabuwono, A. S. (2023). Educational innovation faced with covid-19: Deep learning for online exam cheating detection. *Education Sciences*, 13(2):194.
- Zhou, T. and Jiao, H. (2022). Data augmentation in machine learning for cheating detection in large-scale assessment. *Psychological Test and Assessment Modeling*, 64(4):425–444.